

# Approximate Shortest Descent Path on a Terrain

Sasanka Roy\*, Sachin Lodha\*, Sandip Das† and Anil Maheshwari‡

## Abstract

A path from a point  $s$  to a point  $t$  on the surface of a polyhedral terrain is said to be *descent* if for every pair of points  $p = (x(p), y(p), z(p))$  and  $q = (x(q), y(q), z(q))$  on the path, if  $\text{dist}(s, p) < \text{dist}(s, q)$  then  $z(p) \geq z(q)$ , where  $\text{dist}(s, p)$  denotes the distance of  $p$  from  $s$  along the aforesaid path. Although an efficient algorithm to decide if there is a descending path between two points is known for more than a decade, no efficient algorithm is yet known to find a shortest descending path from  $s$  to  $t$  in a polyhedral terrain. In this paper we propose an  $(1 + \epsilon)$ -approximation algorithm running in polynomial time for the same.

## 1 Introduction

The problem of finding *descending paths* in a polyhedral terrain was first studied by Berg and Kreveld [4]. They presented an  $O(n \log n)$  time algorithm to decide if there is a descending path between two points where  $n$  is the number of faces of the triangulated terrain. Roy et al. [6] presented an  $O(n^2 \log n)$  time algorithm to compute a shortest descending path (SDP) in a convex terrain, and an  $O(n \log n)$  time algorithm to compute an SDP through a sequence of parallel edges. Recently, Ahmed and Lubiw [1] have proposed an  $(1 + \epsilon)$ -approximation algorithm for SDP over a given face sequence using convex optimization techniques. The running time of their algorithm is  $O(n^{3.5} \log(\frac{1}{\epsilon}))$ . Motivation for studying this problem is discussed in [1, 4, 6].

In this paper we propose an  $(1 + \epsilon)$ -approximation algorithm for SDP in a polyhedral terrain. The time complexity of our proposed algorithm is  $O(mn \log mn + nm^2)$  time, where  $n$  is the number of faces of the triangulated terrain and  $m$  is approximately  $O(\frac{n}{\epsilon} \log \frac{1}{\epsilon})$  ignoring the geometric parameters (see Claim 2). Thus we are able to provide the first polynomial time approximation algorithm for the open question raised in [4]. We make use of the discretization method of Aleksandrov et al. [3] that was used to solve the shortest path problem in a weighted terrain. The method involves insert-

ing Steiner points on the edges of the terrain and then constructing an edge weighted graph  $G$ . Finally, the problem boils down to finding a shortest path between two points in  $G$ .

One of the reviewers of this paper has pointed out a very recent technical report [2] by Ahmed and Lubiw that proposes an  $(1 + \epsilon)$ -approximation algorithm for SDP in a polyhedral terrain. It discretizes the terrain with  $O(n^2 \frac{X}{\epsilon})$  Steiner points so that after an  $O(n^2 \frac{X}{\epsilon} \log(\frac{nX}{\epsilon}))$ -time preprocessing phase for a given vertex  $s$ , it can report an  $(1 + \epsilon)$ -approximate path from  $s$  to any point  $v$  in  $O(n)$  time if  $v$  is either a vertex of a terrain or a Steiner point, and in  $O(\frac{nX}{\epsilon})$  time otherwise. Note that  $X = \frac{L}{h} \sec \theta$ , where  $L$  is the length of the longest edge,  $h$  is the smallest distance of a vertex from a non-adjacent edge in the same face, and  $\theta$  is the largest acute angle between a non-level edge and a perpendicular line.

## 2 Preliminaries

A terrain  $\mathcal{T}$  is a polyhedral surface in  $\mathbb{R}^3$  with a special property: the vertical line at any point on the  $xy$ -plane intersects the surface of  $\mathcal{T}$  at most once. Thus, the projections of all the faces of a terrain on the  $xy$ -plane are mutually non-intersecting at their interior. Each vertex  $p$  of the terrain is specified by a triple  $(x(p), y(p), z(p))$ . Without loss of generality, we assume that all the faces of  $\mathcal{T}$  are triangles, and the source  $s$  and destination  $t$  are the vertices of the terrain. Our aim is to find an  $(1 + \epsilon)$ -approximate SDP in  $\mathcal{T}$  from  $s$  to  $t$ . Our algorithm works in three phases.

**Phase 1:** In the first phase, we find out the descent flow region for the source  $s$  using the method that was described in [6]. Given an arbitrary point  $p$  on the surface of the terrain  $\mathcal{T}$ , the descent flow region of  $p$  (called  $DFR(p)$ ) is the region on the surface of  $\mathcal{T}$  such that each point  $q \in DFR(p)$  is reachable from  $p$  through a descent path. Then we check whether  $t$  lies inside  $DFR(s)$  is true. If the answer is negative, then descent path from  $s$  to  $t$  does not exist. If the answer is positive, then we proceed to phase 2. Note that, like  $\mathcal{T}$ ,  $DFR(s)$  is also triangulated.

**Phase 2:** In this phase we use a method similar to the one that was employed by Aleksandrov et al. [3] and construct a graph  $G$ . We deviate from [3] on two accounts: We insert two sets of Steiner points on the

\*Tata Consultancy Services Ltd., Pune, India, sasanka.roy@tcs.com, sachin.lodha@tcs.com

†Indian Statistical Institute, Kolkata, India, sandipdas@isical.ac.in

‡School of Computer Science, Carleton University, Canada, anil@scs.carleton.ca

edges of the  $\mathcal{T}$  instead of one. Of these two sets, the first set of Steiner points is same as that of [3]. The second difference is that, unlike in [3], the graph edges are directed. We discuss these details in the next section.

**Phase 3:** We use Dijkstra's algorithm [5] using Fibonacci heaps to find the shortest path between  $s$  and  $t$  in the directed graph  $G$ . We later show that this path is, in fact, an  $(1 + \epsilon)$ -approximation to SDP.

### 3 Graph $G$

#### 3.1 Steiner Points Insertion

We insert two different sets of Steiner points on the edges of the  $DFR(s)$ .

**$\epsilon$ -Steiner-points:** This set of Steiner points is the same as that of [3]. Let  $v$  be a vertex of  $DFR(s)$ . Define  $h_v$  to be the minimum distance from  $v$  to the boundary of the union of its incident faces. Let  $r_v = \epsilon h_v$  for some  $\epsilon > 0$ .

For each vertex  $v$  of face  $f_i$ , do the following: Let  $e_q$  and  $e_p$  be the edges of  $f_i$  incident to  $v$ . First, place Steiner points on the edges  $e_q$  and  $e_p$  at distance  $r_v$  from  $v$ ; call them  $q_1$  and  $p_1$ , respectively. By definition,  $|vq_1| = |vp_1| = r_v$ . Let  $\theta_v$  be the angle between  $e_p$  and  $e_q$ . Define

$$\delta = \begin{cases} (1 + \epsilon \cdot \sin \theta_v) & \text{if } \theta_v < \frac{\pi}{2}, \\ \delta = (1 + \epsilon) & \text{otherwise.} \end{cases}$$

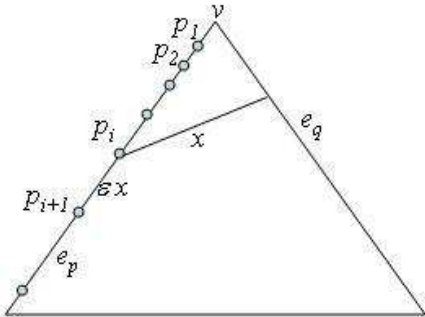


Figure 1:  $\epsilon$ -Steiner Points

We now add Steiner points  $q_2, q_3, \dots, q_{\mu_q-1}$  along  $e_q$  such that  $|vq_j| = r_v \delta^{j-1}$  where  $\mu_q = \lceil \log_\delta(\frac{|e_q|}{r_v}) \rceil$ . Similarly, add Steiner  $p_2, p_3, \dots, p_{\mu_p-1}$  along  $e_p$  where  $\mu_p = \lceil \log_\delta(\frac{|e_p|}{r_v}) \rceil$ . See Figure 1.

Define  $dist(a, e)$  as the minimum distance from a point  $a$  to an edge  $e$ . This segment from  $a$  to  $e$  will be perpendicular to  $e$ .

**Claim 1** (3.11 of [3])  $|q_i q_{i+1}| \leq \epsilon \cdot dist(q, e_p)$  and  $p_j p_{j+1} \leq \epsilon \cdot dist(p, e_q)$  where  $0 < i < \mu_q$ ,  $0 < j < \mu_p$ ,  $q \in q_i q_{i+1}$  and  $p \in p_j p_{j+1}$ .

**Isohypse-Steiner-points:** Let  $p_1, p_2, \dots, p_{\mu_q}$  be the set of  $\epsilon$ -Steiner-points on some edge  $e_p$ . For any non-horizontal edge (An edge  $e$  is horizontal if for any two points  $a$  and  $b$  on the edge  $e$ ,  $z(a) = z(b)$ )  $e_j (\neq e_i)$  add an Isohypse-Steiner-point  $dp_i$  on edge  $e_j$  if  $z(dp_i) = z(p_i)$ . So, if there is no point on edge  $e_j$  such that  $z(dp_i) = z(p_i)$ , then no Isohypse-Steiner-point is inserted. Intuitively, we take a horizontal plane at each Steiner point and intersect it with the terrain. At each intersection of the plane and an edge of terrain we insert a Steiner point. Note that, the insertion of Isohypse-Steiner-point may increase the total number of  $\epsilon$ -Steiner-points by at most a factor of  $n$ .

#### 3.2 Graph Construction

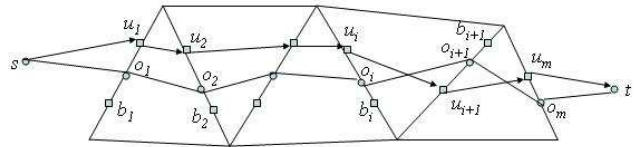


Figure 2: Approximation of SDP

For each face  $f_i$  we treat all the  $\epsilon$ -Steiner-points, Isohypse-Steiner-points and the vertices of the  $DFR(s)$  as the node of graph  $G_i$ . Two vertices  $a$  and  $b$  of  $G_i$  are connected by a directed edge  $e_{ab}$  if  $a$  and  $b$  lie on two different edges of face  $f_i$  and  $z(a) \geq z(b)$ . The weight of each edge is the Euclidean distance between  $a$  and  $b$ . Now we define an edge weighted directed graph  $G = G_1 \cup G_2 \cup \dots \cup G_n$ .

**Claim 2** At most  $m = O(n \log_\delta(\frac{|L|}{r}))$  Steiner points are added to each edge of  $f_i$ , for  $1 \leq i \leq n$ , and where  $|L|$  is the length of the longest edge in  $DFR(s)$  and  $r$  is the minimum among the  $r_v$ .

**Claim 3**  $G$  has  $O(n^2 \log_\delta(\frac{|L|}{r}))$  vertices and  $O(n^3 \log_\delta^2(\frac{|L|}{r}))$  edges.

### 4 Proof of Approximation Factor

Let  $\pi(s, t) = [s, o_1, o_2, \dots, o_m, t]$  be the shortest monotone descent path and it passes through the edge sequence  $e_1, e_2, \dots, e_m$ . Then in each edge  $e_i$  the path  $\pi(s, t)$  must pass through a pair of Steiner-points which are closest to  $o_i$ , say  $u_i$  and  $b_i$  (See Figure 2) such that  $z(u_i) \geq z(o_i) \geq z(b_i)$ . Note that, in degenerate case point  $u_i$  may be  $o_i$  and  $o_i$  may coincide with  $b_i$ . Let us consider the path  $\pi^*(s, t) = [s, u_1, u_2, \dots, u_m, t]$ .

**Lemma 4**  $\pi^*(s, t)$  is monotone descent.

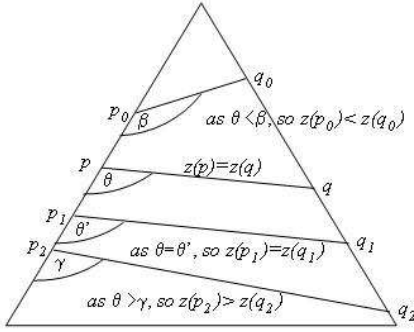


Figure 3: Illustration of Lemma 5 and Corollary 6

To prove the above Lemma we need to prove the following statement.

**Lemma 5** Let  $p$  and  $q$  be two points on two consecutive members  $e_i$  and  $e_{i+1}$  of  $\mathcal{E}$  which bounds a face  $f$ , and  $z(p) = z(q)$ . Now, if a line  $\ell$  on face  $f$  intersects both  $e_i$  and  $e_{i+1}$ , and is parallel to the line segment  $[p, q]$ , then all the points on  $\ell$  have the same  $z$ -coordinate.

**Proof.** Consider a horizontal plane  $h$  at altitude  $z(p)$ . The intersection of the face  $f$  and the plane  $h$  is the line segment  $[p, q]$ . Consider another horizontal plane  $h'$  through a point  $r$  on line  $\ell$ . The intersection of  $f$  and  $h'$  must be line parallel to  $[p, q]$ , and hence it coincides with the line  $\ell$ . Thus, all the points on the line  $\ell$  have the same  $z$ -coordinate.  $\square$

From the above Lemma, we get the following Corollary.

**Corollary 6** If  $[p, q]$  makes an angle  $\theta$  with  $e_i$  (in anticlockwise direction see Figure 3) and  $[v, w]$  ( $v \in e_i$  and  $w \in e_{i+1}$ ) be the line segment that makes an angle  $\beta$  with  $e_i$  (in anticlockwise direction). Now

1.  $\theta = \beta$  iff  $z(v) = z(w)$ ,
2.  $\theta > \beta$  iff  $z(v) > z(w)$ , and
3.  $\theta < \beta$  iff  $z(v) < z(w)$ .

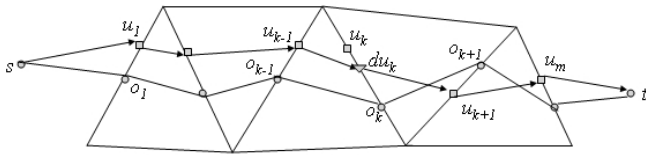


Figure 4: Proof of Lemma 4

Now we have to prove that the path  $\pi^*(s, t)$  is monotone descent. Without loss of generality we can say that

$z(s) \geq z(u_1)$ . We will prove this using contradiction. So the path from  $s$  to  $u_1$  is monotone descent. Let us consider that path  $s$  to  $u_{k-1}$  along  $\pi'(s, t)$  is monotone descent and the path segment  $u_{k-1}$  to  $u_k$  is not monotone descent. Then  $z(u_{k-1}) < z(u_k)$ . Again the path segment  $o_{k-1}$  to  $o_k$  is monotone descent and  $z(u_k) > z(o_k)$  (As  $z(u_k) > z(b_k)$ ). So there is a Isohypse-Steiner-point  $du_k \in [u_k, o_k]$  (by Lemma 5 and Corollary 6 see Figure 4) such that path from  $s$  to  $du_k$  is monotone descent. This contradicts the assumption that  $u_k$  is closet to  $o_k$ .

From the above discussion, we can conclude that the path  $\pi^*(s, t)$  is monotone descent. Hence Lemma 4 follows.

**Lemma 7**  $|\pi^*(s, t)| \leq (1 + 2\epsilon)|\pi(s, t)|$ , where  $|\pi(a, b)|$  implies the length of the path  $\pi(a, b)$ .

**Proof.** We have

$$\begin{aligned}
 |\pi^*(s, t)| &= |su_1| + |u_1u_2| + |u_2u_3| + \dots + \\
 &\quad + |u_{m-1}u_m| + |u_mt| \\
 &\leq |so_1| + |o_1u_1| + |u_1o_1| + |o_1o_2| + \\
 &\quad + |o_2u_2| + |u_2o_2| + |o_2o_3| + \\
 &\quad + |o_3u_3| + |u_3o_3| + \dots + |o_{m-1}u_{m-1}| \\
 &\quad + |o_{m-1}o_m| + |o_mu_m| + |u_m o_m| + |o_mt| \\
 &= |so_1| + |o_1o_2| + |o_2o_3| + \dots + \\
 &\quad + |o_{m-1}o_m| + |o_mt| + \\
 &\quad + 2\{|o_1u_1| + |o_2u_2| + \dots + |u_m o_m|\} \\
 &\leq |so_1| + |o_1o_2| + |o_2o_3| + \dots + \\
 &\quad + |o_{m-1}o_m| + |o_mt| + \\
 &\quad + 2\epsilon\{|o_1o_2| + |o_2o_3| + \dots + |o_{m-1}o_m|\} \\
 &\quad \text{(using Claim 1, Lemma 5 and Corollary 6)} \\
 &\leq (1 + 2\epsilon)\pi(s, t).
 \end{aligned}$$

$\square$

From the above Lemma we can conclude that there exists a path  $\pi^*(s, t)$  such that  $|\pi^*(s, t)| = (1 + 2\epsilon)|\pi(s, t)|$ . Dijkstra's algorithm may output a monotone descent path  $\pi^{**}(s, t)$  which may be a different path from  $\pi^*(s, t)$ . As Dijkstra's algorithm outputs a path of shortest length in graph  $G$ , we have  $|\pi^*(s, t)| \geq |\pi^{**}(s, t)|$ . So, we can conclude the following theorem.

**Theorem 8** Let  $0 < \epsilon < \frac{1}{2}$ . Let  $DFR(s)$  be a terrain with  $n$  faces and let  $s$  and  $t$  be two of its vertices. An approximation  $\pi'(s, t)$  of a shortest monotone path  $\pi(s, t)$  can be computed such that  $|\pi'(s, t)| \leq (1 + 2\epsilon)|\pi(s, t)|$ . The approximation can be computed in  $O(mn \log mn + nm^2)$  time where  $m = O(n \log_\delta(\frac{L|l|}{r}))$  and  $\delta = (1 + \epsilon \sin \theta)$  (by virtue of Claim 3), when  $\theta, L, r$  denote the smallest angle among the triangles, the length of the longest edge, and the length of smallest  $r_v$ , respectively.

**Proof.** The approximation factor follows from Lemma 7. The running time of the algorithm is same as that of single source shortest path algorithm on a graph with  $O(n^2 \log_\delta(\frac{|L|}{r}))$  vertices and  $O(n^3 \log_\delta^2(\frac{|L|}{r}))$  edges.  $\square$

## References

- [1] M. Ahmed and A. Lubiw, Shortest descending paths through given faces, *Proc. Canadian Conference on Computational Geometry*, pp. 35-38, 2006.
- [2] M. Ahmed and A. Lubiw, An Approximation Algorithm for Shortest Descending Paths, *University of Waterloo Technical Report CS-2007-14*, May 2007, <http://arxiv.org/abs/0705.1364v1>.
- [3] L. Aleksandrov, M. Lanthier, A. Maheshwari and J. R. Sack, An  $\epsilon$ -Approximation for Weighted Shortest Paths on Polyhedral Surfaces, *Proc. Scandinavian Workshop on Algorithm Theory*, pp. 11-22, 1998.
- [4] M. de Berg and M. van Kreveld, Trekking in the Alps without freezing or getting tired, *Algorithmica*, vol. 18, pp. 306-323, 1997.
- [5] E.W. Dijkstra, A Note on Two Problems in Connection with Graphs, *Numerical Mathematics*, vol. 1, pp. 269-271, 1959.
- [6] S. Roy, S. Das and S. C. Nandy, Shortest monotone descent path problem in polyhedral terrain, *Proc. Symp. on Theoretical Aspects of Computer Science*, pp. 281-292, 2005.