

# Shortest Gently Descending Paths

Mustaq Ahmed<sup>1</sup>, Anna Lubiw<sup>1\*</sup>, and Anil Maheshwari<sup>2\*</sup>

<sup>1</sup> D. R. C. School of Computer Science, University of Waterloo, ON, Canada.  
Email: m6ahmed@uwaterloo.ca, alubiw@uwaterloo.ca

<sup>2</sup> School of Computer Science, Carleton University, ON, Canada.  
Email: anil@scs.carleton.ca

**Abstract.** A path from  $s$  to  $t$  on a polyhedral terrain is *descending* if the height of a point  $p$  never increases while we move  $p$  along the path from  $s$  to  $t$ . We introduce a generalization of the shortest descending path problem, called the *shortest gently descending path (SGDP)* problem, where a path descends, but not too steeply. The additional constraint to disallow a very steep descent makes the paths more realistic in practice. We give two approximation algorithms (more precisely, FPTASs) to solve the SGDP problem on general terrains.

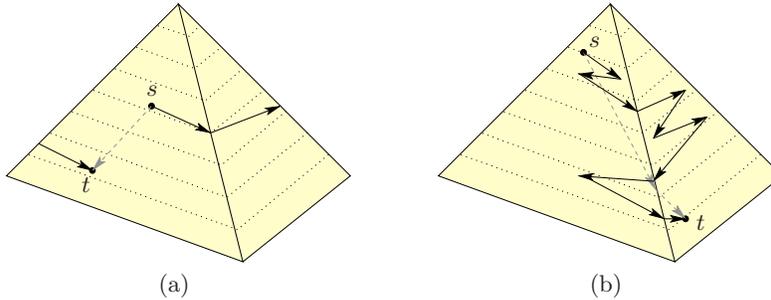
## 1 Introduction

A well-studied problem in computational geometry is to compute a shortest path on a polyhedral terrain. One variant of the problem for which good approximation algorithms have recently been found [3, 14] is the *shortest descending path (SDP) problem*: given a polyhedral terrain, and points  $s$  and  $t$  on the surface, find a shortest path on the surface from  $s$  to  $t$  such that, as a point travels along the path, its elevation, or  $z$ -coordinate, never increases. In many applications of SDPs, we want a path that descends, but *not too steeply*. For example, when we ski down a mountain we avoid a too steep descent. In such cases, a very steep segment of a descending path should be replaced by “switchbacks” that go back and forth at a gentler slope, like the hairpin bends on a mountain road (Figure 1). The *shortest gently descending path (SGDP)* problem combines two previously-studied problems: (i) to find SDPs; and (ii) to find shortest *anisotropic* paths, where there are different costs associated with traveling in different directions in a face. Our constraint that forbids a steep descent is an anisotropic constraint. At first glance it would seem that the entire SGDP problem is a special case of anisotropic paths, but results on anisotropic paths assume (e.g., Sun and Reif [17]) that there is a feasible path between any two points in a common face, a property that is violated when ascending paths are forbidden.

In this paper we combine techniques used for SDPs and for anisotropic paths and present two fully polynomial time approximation schemes (FPTASs) to solve the SGDP problem on a general terrain. We model the problem as a shortest path problem in a graph whose nodes are Steiner points added along the edges

---

\* Research supported by NSERC



**Fig. 1.** Descending gently towards a steep direction.

of the terrain, with directed edges from higher to lower points in a common face, and edge weights corresponding to gently descending distances. Both the algorithms are simple, robust and easy to implement. We measure performance in terms of the number  $n$  of vertices of the terrain, the largest degree  $d$  of a vertex, the desired approximation factor  $\epsilon$ , and a parameter  $X$  that depends on the geometry of the terrain and the measure of steepness (see Section 3). In our first algorithm, given a vertex  $s$ , we place Steiner points uniformly along terrain edges during an  $O\left(\frac{n^2 X}{\epsilon} \log\left(\frac{n X}{\epsilon}\right)\right)$ -time preprocessing so that we can determine a  $(1 + \epsilon)$ -approximate SDP from  $s$  to any point  $v$  in  $O(nd)$  time if  $v$  is either a vertex of the terrain or a Steiner point, and in  $O\left(n\left(d + \frac{X}{\epsilon}\right)\right)$  time otherwise. Our second algorithm places Steiner points in geometric progression along the edges to make the algorithm less dependent (than the first algorithm) on the slopes of the edges at the cost of slightly more dependency on  $n$ , see Theorem 2 for details. We can easily combine these two algorithms into a “hybrid” one: first check the edge inclinations of the input terrain, and then run whichever of these two algorithms ensures a better running time for that particular terrain.

The paper is organized as follows. In Sections 2 and 3 we mention related results and define a few terms. Section 4 establishes a few properties of an SGDP. Sections 5 and 6 give our approximation algorithms.

## 2 Related Work

The SDP problem was introduced by de Berg and van Kreveld [8], who gave an  $O(n \log n)$  time algorithm to decide existence of a descending path between two points. Until recently the SDP problem has been studied in different restricted settings [2, 13]. Two recent papers [3, 14] give approximation algorithms for the problem on general terrains. Both papers use the Steiner point approach, i.e., the approach of discretizing the continuous space by adding Steiner points and approximating a shortest path through the space by a shortest path in the graph of Steiner points. Details of these algorithms appear in Ahmed et al. [1]. Another recent work on SDP [4] gives a full characterization of the bend angles

of an SDP, and points out the difficulty of finding an exact SDP on a general terrain.

The Steiner point approach has been used for other variants of shortest paths on terrains. One of them is the weighted region problem [11]. See Aleksandrov et al. [6] for a brief survey of Steiner point algorithms for the problem, and Sun and Reif [18] for more recent work. One generalization of the Weighted Region Problem is finding a shortest anisotropic path [12], where the weight assigned to a region depends on the direction of travel. The weights in this problem capture, for example, the effect the gravity and friction on a vehicle moving on a slope. All the papers on this problem use the Steiner point approach [7, 10, 15, 17]. As we mentioned before, these algorithms for anisotropic paths assume that every face  $f$  is *totally traversable*, i.e., there is a feasible path from any point to any other point in  $f$ . To be precise, Cheng et al. [7] assumes that the (anisotropic) weight associated with a direction of travel is bounded by constants from both above and below, thus any direction of travel is feasible. (Moreover, the algorithm of Cheng et al. is for a planar subdivision, not for a terrain.) The rest of the papers [10, 15, 17] use the anisotropic weight model of Rowe and Ross [12] which allows switchback paths to “cover” any direction in  $f$ . The assumption that every face is totally traversable allows placing Steiner points in a face *independently from all other faces*. Sun and Reif [17, Section V] relax this assumption (i.e. the assumption that every face is totally traversable) but only in isolated faces. Thus, they can still rely on independent placement of Steiner points in a face. For both the SDP and the SGDP problems, ascending directions are unreachable in *every* face, which necessitates the use of a non-local strategy of placing Steiner points.

To obtain a better running time our algorithms use a variant of Dijkstra’s algorithm, called the Bushwhack algorithm [16], to compute a shortest path in the graph of Steiner points. In such a graph, the Bushwhack algorithm improves the running time of Dijkstra’s algorithm from  $O(|V| \log |V| + |E|)$  to  $O(|V| \log |V|)$ .

### 3 Terminology

A terrain is a 2D surface in 3D space with the property that every vertical line intersects it in at most one point. For any point  $p$  in the terrain,  $h(p)$  denotes the height of  $p$ , i.e., the  $z$ -coordinate of  $p$ . We consider a triangulated terrain, and add  $s$  as a vertex. The terrain has  $n$  vertices, and hence at most  $3n$  edges and  $2n$  faces by Euler’s formula [9]. Let  $L$  be the length of the longest edge,  $h$  be the smallest distance of a vertex from a non-adjacent edge in the same face (i.e. the smallest 2D height of a triangular face),  $d$  be the largest degree of a vertex, and  $\theta$  be the largest acute angle between a non-level edge and a vertical line.

In this paper, “edge” and “vertex” denote respectively a line segment of the terrain and an endpoint of an edge, “segment” and “node” denote respectively a line segment of a path and an endpoint of a segment, and “node” and “link” denote the corresponding entities in a graph. We assume that all paths are directed. In our figures dotted lines denote level lines.

A path  $P$  from  $s$  to  $t$  on the terrain is *descending* if the  $z$ -coordinate of a point  $p$  never increases while we move  $p$  along the path from  $s$  to  $t$ . Given an angle  $\psi \in [0, \frac{\pi}{2})$ , a line segment  $pq$  is *steep* if it makes an angle less than  $\psi$  with a vertical line. A path  $P$  is *gently descending* if  $P$  is descending, and no segment of  $P$  is steep. A downward direction in a face is called a *critical direction* if the direction makes an angle equal to  $\psi$  with a vertical line. A gently descending path is called a *critical path* if each of its segments is in a critical direction. A critical path may travel through more than one face; inside a face it will zig-zag back and forth. We would like to replace steep descending segments by critical paths. This is sometimes possible, e.g. for a steep segment starting and ending at points interior to a face, but is not possible in general. The details are in Lemma 3, which uses the following terms. A vertex  $v$  in face  $f$  is *locally sharp in  $f$*  if  $v$  is either the higher endpoint of two steep edges or the lower endpoint of two steep edges of  $f$ . A vertex  $v$  is *sharp* if it is locally sharp in all its incident faces. Note that a sharp vertex is either the higher endpoint of all the edges incident to it, or the lower endpoint of all such edges. A sharp vertex is like a pinnacle from which you cannot descend gently.

## 4 Properties of an SGDP

Because our approximation algorithms use the Bushwhack algorithm which relies on the optimality of subpaths of a shortest path, we need to establish a similar property of an SGDP:

**Lemma 1.** *Any subpath of an SGDP is an SGDP.*

Another property that is crucial for our algorithm (and perhaps for *any* SGDP algorithm) is that any critical path in the terrain is an SGDP:

**Lemma 2.** *Any critical path from a point  $a$  to a point  $b$  in the terrain is an SGDP of length  $(h(a) - h(b)) \sec \psi$ .*

*Proof.* Any critical path  $P$  is a gently descending path. Since each segment of  $P$  makes an angle  $\psi$  with a vertical line, the length of  $P$  is  $(h(a) - h(b)) \sec \psi$ . Ignoring the terrain, *any* gently descending path from  $a$  to any point at height  $h(b)$  has length at least  $(h(a) - h(b)) \sec \psi$ . So,  $P$  is an SGDP.  $\square$

We will now define the following notation to simplify our expressions involving the length of an SGDP. For any two points  $p$  and  $q$  in a common  $f$ , let  $\|pq\| = |(h(p) - h(q)) \sec \psi|$  when line  $pq$  is steep, and  $\|pq\| = |pq|$  otherwise. Thus  $\|pq\|$  is the length of an SGDP from  $p$  to  $q$  if one exists.

**Observation 1.** *For any three points  $p$ ,  $q$  and  $r$  in a face  $f$ : (i)  $\|pq\| = \|qp\|$ , (ii)  $\|pr\| \leq \|pq\| + \|qr\|$ , and (iii)  $|pq| \leq \|pq\| \leq |pq| \sec \psi$ .*

*Proof.* The proof is obvious except for the second inequality of Case (iii), which follows from the inequalities  $\sec \psi \geq 1$  and  $|pq| \geq |h(p) - h(q)|$ .  $\square$

Like other Steiner point approaches, our graph of Steiner points has a directed link (of appropriate cost) between two Steiner points  $a$  and  $b$  in a common face  $f$  such that the link represents an SGDP from  $a$  to  $b$ . However, unlike other well-studied shortest paths in terrains (e.g., SDPs and shortest paths in the weighted region problem) where the shortest path represented by the link lies completely in  $f$ , the SGDP in our case may go through many other faces. For example, the critical path in Figure 1(a) is an SGDP by Lemma 2, and it goes through four faces even though both  $s$  and  $t$  lie on a common face. It is not straightforward to determine if such an SGDP exists at all—we need this information during the construction of the graph. Moreover, in case an SGDP exists, we want to know the number of faces used by the path because our algorithm has to return the path in the terrain that corresponds to the shortest path in the graph. Lemma 3 below handles these issues:

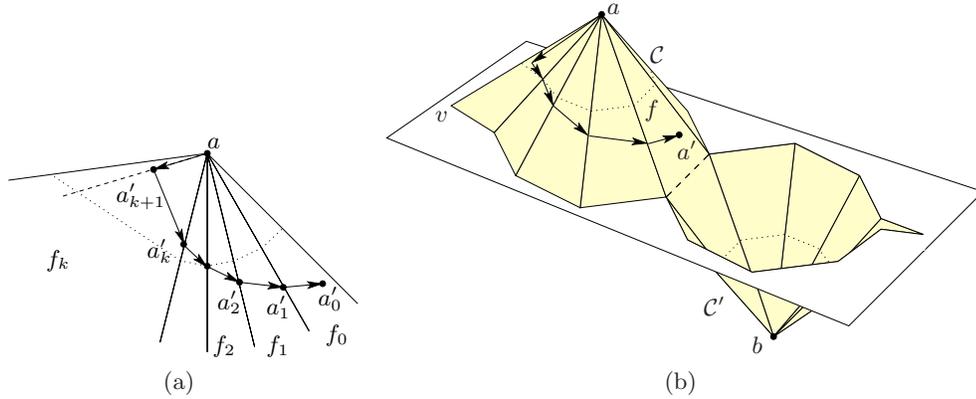
**Lemma 3.** *Let  $a$  and  $b$  be two points in a face  $f$  with  $h(a) \geq h(b)$ .*

- (i) *If any of  $a$  and  $b$  is a sharp vertex, no gently descending path exists from  $a$  to  $b$ .*
- (ii) *If neither  $a$  nor  $b$  is a locally sharp vertex in  $f$ , there exists an SGDP from  $a$  to  $b$  lying completely in  $f$ . Moreover, the SGDP is a critical path if  $ab$  is steep.*
- (iii) *Otherwise, there exists an SGDP from  $a$  to  $b$  that uses at most  $d + 1$  faces, and is a critical path.*

*Proof.* (i) If  $a$  [or  $b$ ] is a sharp vertex, the segment  $ap$  [respectively  $pb$ ] is steep for any point  $p$  in any face incident to  $a$  [respectively  $b$ ]. So, no gently descending path exists from  $a$  to  $b$ .

(ii) If  $ab$  is not a steep segment, this is the SGDP. If  $ab$  is a steep segment, observe that there is a point  $a' \in f$  such that  $aa'$  is a critical path, and  $a'b$  is a steep segment. Moreover,  $h(a') < h(a)$ , and therefore,  $\|a'b\| < \|ab\|$  by definition. Thus by going from  $a$  to  $a'$ , we move closer (in terms of gently descending distances) to  $b$  along a critical direction from  $a$ . We can use the same argument for a critical direction from  $a'$  since  $a'b$  is steep. By repeating this argument, we get a critical path from  $a$  to  $b$ , which is an SGDP by Lemma 2.

(iii) In this case, at least one of  $a$  and  $b$  is a locally sharp vertex in  $f$ , and therefore,  $ab$  is a steep line segment. If  $a$  is not a locally sharp vertex in  $f$ , let  $a' = a$ . Otherwise, let  $a'$  be an interior point of line segment  $ab$  such that no vertex of the terrain lies strictly in between the planes  $z = h(a)$  and  $z = h(a')$ . We similarly define a point  $b'$ , and make sure that  $h(a') > h(b')$ . Clearly  $h(a) > h(a') > h(b') > h(b)$ , and  $a'b'$  is a steep line segment. By Case (ii) of the lemma there exists a critical path from  $a'$  to  $b'$  in  $f$ . We claim that there exists a critical path from  $a$  to  $a'$  through at most  $\lfloor \frac{d}{2} \rfloor + 1$  faces when  $a \neq a'$ , and that there exists a critical path from  $b'$  to  $b$  through at most  $\lfloor \frac{d}{2} \rfloor + 1$  faces when  $b \neq b'$ . Because face  $f$  is used by all these three subpaths, the whole path uses at most  $d + 1$  faces. The proof then follows



**Fig. 2.** (a) A critical path from a vertex  $a$  that is locally sharp in  $f = f_0$ , but not in  $f_k$ . (b) An SGDP between two locally sharp vertices  $a$  and  $b$  in  $f$  that goes through  $\Theta(d)$  other faces.

from Lemma 2. We will now prove the first claim. The proof for the second claim is similar, and hence omitted.

Since  $a$  is not a sharp vertex, it must have some incident face in which it is not locally sharp. Let  $(f = f_0, f_1, f_2, \dots, f_k)$  be (one of) the shortest sequence of faces around vertex  $a$  such that  $a$  is not a locally sharp vertex in  $f_k$  (Figure 2(a)). Clearly,  $k \leq \lfloor \frac{d}{2} \rfloor + 1$ . We will build as follows a critical path backwards from  $a'$ . Let  $a'_0 = a'$ . For each  $i \in [0, k-1]$  in this order, since  $a$  is a locally sharp vertex in  $f_i$ , there is a point  $a'_{i+1} \in f_i \cap f_{i+1}$  such that  $a'_{i+1}a'_i$  is a critical direction in  $f_i$ . The final point  $a_k$  lies on the edge between  $f_k$  and  $f_{k-1}$  which is a steep edge. Because  $a$  is not a locally sharp vertex in  $f_k$ , we can ensure by making  $a'$  sufficiently close to  $a$  that there exists an interior point  $a'_{k+1} \in f_k$  such that both  $aa'_{k+1}$  and  $a'_{k+1}a'_k$  are critical directions. (If  $a'$  is not sufficiently close to  $a$ , there still exists a critical path from  $a$  to  $a'_k$  but the path may have more than one bends.) Clearly the path  $(a, a'_{k+1}, a'_k, \dots, a'_0 = a')$  is a critical path through at most  $\lfloor \frac{d}{2} \rfloor + 1$  faces.  $\square$

It is easy to construct a terrain in which the number of faces in Case (iii) of the above lemma is exactly  $d$ , see Figure 2(b) for an example. In fact, we can prove that such a path through  $d+1$  faces is impossible. We omit the proof here.

Another property of an SGDP we would like to utilize is that an SGDP visits a face at most once, because our method of approximating a path introduces some error each time the path crosses an edge. But unlike SDPs and shortest paths in the weighted region problem, this property does *not* hold for an SGDP. For example, the two SGDPs in Figure 1 visit a face twice. We can even make the SGDP in Figure 1(b) visit a face infinitely many times, e.g., by making angle  $\psi$  arbitrarily close to  $\frac{\pi}{2}$  so that the path spirals around the pyramid. We call an SGDP *ideal* if it crosses the interior of each face at most once. The good news

is that we can “convert” any non-ideal SGDP into an ideal one—we will prove this claim in the following lemma:

**Lemma 4.** *If there is a gently descending path from  $s$  to  $t$ , there exists an ideal SGDP from  $s$  to  $t$ .*

*Proof.* Let  $P$  be an SGDP from  $s$  to  $t$ . Such a path exists because one gently descending path from  $s$  to  $t$  must be the shortest one. It suffices to show that if  $P$  visits the interior of a face  $f$  more than once, we can replace the portion between the first and the last visit by a shortcut that is gently descending, remains in face  $f$ , and has length no greater than the original. Let  $P_a$  [ $P_b$ ] be the first [respectively last] path in  $P \cap f$  that crosses the interior of face  $f$ . Let  $a$  [ $b$ ] be the first [respectively last] point of  $P_a$  [respectively  $P_b$ ]. Note that  $a$  and  $b$  are not locally sharp vertices in  $f$ . If segment  $ab$  is not steep, then it provides the shortcut. If segment  $ab$  is steep, then by Lemma 3(ii) there is an SGDP from  $a$  to  $b$  lying inside  $f$ .  $\square$

## 5 Approximation using Uniform Steiner Points

### 5.1 Algorithm

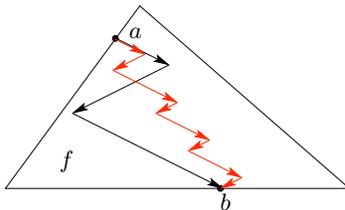
In our first approximation algorithm, we preprocess the terrain by adding uniformly spaced Steiner points as follows. We take a set of level planes such that the distance between any two consecutive planes is at most  $\delta = \frac{\epsilon h}{4n} \cos \theta \cos \psi$ . We make sure that there is a level plane through every vertex. We then put Steiner points at all the intersection points of these planes with the non-level edges of the terrain. On the level edges, we add enough Steiner points so that consecutive points are at most  $\delta \sec \theta$  units apart. We then construct a weighted directed graph  $G(V, E)$  in which each node in  $V$  represents either a Steiner point or a vertex, and there is a directed link  $pq \in E$  if and only if all of the following are true:

- (i)  $p$  and  $q$  lie in a common face,
- (ii)  $h(p) \geq h(q)$ , and
- (iii) neither  $p$  nor  $q$  is a sharp vertex.

By Lemma 3, there is an SGDP from  $p$  to  $q$ . The weight of link  $pq$  is the length of such an SGDP, i.e.,  $\|pq\|$ . In the final step of our preprocessing we make a shortest path tree  $T$  rooted at  $s$  using the Bushwhack algorithm. The Bushwhack algorithm works for any distance metric that satisfies the following property: if  $e$  and  $e'$  are two edges of one face, and  $a$  and  $b$  are two points on edge  $e$ , then edge  $e'$  can be divided into two sub-intervals  $A$  and  $B$ , where points in  $A$  have a shorter path from  $a$  than from  $b$  and points in  $B$  have a shorter path from  $b$  than from  $a$ . This property holds for our distance metric, even though an SGDP connecting two points in face  $f$  may leave  $f$ . The proof follows from Sun and Reif [17], Lemmas 3 and 4, where they prove that the property holds for anisotropic paths.

Note that we are mentioning set  $E$  only to make the discussion easy. In practice, we do not construct  $E$  explicitly because the neighbors of a node  $x \in V$  in the graph is determined *during* the execution of the Bushwhack algorithm.

To answer a query, we simply return the path from  $s$  to query point  $v$  in  $T$  if  $v \in V$  and  $v$  is not a sharp vertex. If  $v$  is a sharp vertex, we return nothing since there is no SGDP from  $s$  to  $v$ . Otherwise,  $v \notin V$ . In this case, we find the node  $u$  among those in  $V$  lying in the face(s) containing  $v$  such that  $h(u) \geq h(v)$ , and the sum of  $\|uv\|$  and the length of the path from  $s$  to  $u$  in  $T$  is minimum. Finally we return the corresponding path from  $s$  to  $v$  as an approximate SGDP.



**Fig. 3.** Two of the infinitely many SGDPs from  $a$  to  $b$  in  $f$ .

There is an important issue regarding the path returned by our algorithm. It is a path in the graph augmented by vertex  $v$ . To obtain an actual path on the terrain we must replace each link  $ab$  in the path by an SGDP of the same length, which is possible by the definition of the links of the graph. Such an SGDP is not unique if  $ab$  is steep, (Figure 3) but it is easy to compute one such path. In the case where neither  $a$  nor  $b$  is locally sharp in their common face  $f$ , we can even compute an SGDP with a minimum number of bends. Note, however, that the problem of minimizing the total number of bends in an SGDP is NP-hard [5].

## 5.2 Correctness and Analysis

For the proof of correctness, we show that an ideal SGDP  $P$  from  $s$  to any point  $v$  in the terrain is approximated by a path  $P'$  such that all the segments of  $P'$ , except possibly the last one, exist in  $G$ . For ease of discussion, we will say that a path  $Q$  in  $G$  *represents* a gently descending path in the terrain if for each directed link  $ab$  of  $Q$  there exists a gently descending path in the terrain from  $a$  to  $b$ .

Let  $\sigma_P = (s = p_0, p_1, p_2, \dots, p_k, v = p_{k+1})$  be an ordered subsequence of the nodes in  $P$  such that (a) for each  $i \in [0, k]$ , the segments in-between  $p_i$  and  $p_{i+1}$  lie in a common face  $f_i$ , and (b) for each  $i \in [0, k-1]$ , the segment exiting from  $p_{i+1}$  does *not* lie in  $f_i$ . Note that  $p_i$  and  $p_{i+1}$  are two different boundary points of face  $f_i$  for all  $i \in [0, k-1]$ , and  $p_k$  and  $p_{k+1}$  are two different points of face  $f_k$  ( $p_{k+1}$  can be an interior point of  $f_k$ ). For all  $i \in [0, k]$ , the part of  $P$  between  $p_i$  and  $p_{i+1}$  remains in  $f_i$ . Let  $e_i$  be an edge of the terrain through  $p_i$  for all  $i \in [1, k]$  ( $e_i$  can be any edge through  $p_i$  if  $p_i$  is a vertex).

We construct a graph path  $P'$  with node sequence  $(s = p'_0, p'_1, p'_2, \dots, p'_k, v = p'_{k+1})$  as follows: for each  $i \in [1, k]$ , let  $p'_i = p_i$  if  $p_i$  is a vertex of the terrain; otherwise, let  $p'_i$  be the nearest point from  $p_i$  in  $V \cap e_i$  such that  $h(p'_i) \geq h(p_i)$ .

**Lemma 5.** *For all  $i \in [0, k]$ ,  $h(p'_i) \geq h(p'_{i+1})$ .*

*Proof.* We first claim that  $h(p'_i) \geq h(p_{i+1})$ . This claim follows from the facts that  $h(p'_i) \geq h(p_i)$  by the definition of  $p'_i$ , and  $h(p_i) \geq h(p_{i+1})$  as  $P$  is descending. Now consider the following two cases:

**Case 1:**  $p'_{i+1} = p_{i+1}$  or  $e_{i+1}$  is a level edge. In this case,  $h(p'_{i+1}) = h(p_{i+1})$ . It follows from the inequality  $h(p'_i) \geq h(p_{i+1})$  that  $h(p'_i) \geq h(p'_{i+1})$ .

**Case 2:**  $p'_{i+1} \neq p_{i+1}$  and  $e_{i+1}$  is a non-level edge. In this case, there is either one or no point in  $e_{i+1}$  at any particular height. Let  $p''_{i+1}$  be the point in  $e_{i+1}$  such that  $h(p''_{i+1}) = h(p'_i)$ , or if no such point exists, let  $p''_{i+1}$  be the upper vertex of  $e_{i+1}$ . In the latter case, we can infer from the inequality  $h(p'_i) \geq h(p_{i+1})$  that  $h(p'_i) > h(p''_{i+1})$ . Therefore we have  $h(p'_i) \geq h(p'_{i+1})$  in both cases. Since  $p''_{i+1} \in V \cap e_{i+1}$ , the definition of  $p'_{i+1}$  implies that  $h(p''_{i+1}) \geq h(p'_{i+1})$ . So,  $h(p'_i) \geq h(p'_{i+1})$ .

Therefore,  $h(p'_i) \geq h(p'_{i+1})$  for all  $i \in [0, k]$ . □

**Lemma 6.** *For all  $i \in [0, k + 1]$ ,  $p'_i$  is not a sharp vertex.*

*Proof.* None of  $p'_0 = s$  and  $p'_{k+1} = v$  are sharp vertices because both the segments  $sp_1$  and  $p_kv$  are gently descending. For each  $i \in [1, k]$ , if  $p'_i$  is a sharp vertex, then  $p'_i$  is either the unique topmost vertex or the unique bottommost vertex in all incident faces. Therefore, either

$$h(p'_{i-1}) < h(p'_i) > h(p'_{i+1}) \text{ ,}$$

or

$$h(p'_{i-1}) > h(p'_i) < h(p'_{i+1}) \text{ .}$$

Both of these are impossible by Lemma 5. So  $p'_i$  is not a sharp vertex. □

**Lemma 7.** *The part of  $P'$  from  $s$  to  $p'_k$  exists in  $G$ . Path  $P'$  represents a gently descending path.*

*Proof.* To show that the part of  $P'$  from  $s$  to  $p'_k$  exists in  $G$ , it is sufficient to prove that  $p'_i p'_{i+1} \in E$  for all  $i \in [0, k - 1]$ , because both  $p'_i$  and  $p'_{i+1}$  are in  $V$  by definition. Since  $p'_i$  and  $p'_{i+1}$  are boundary points of face  $f_i$  by definition, and  $h(p'_i) \geq h(p'_{i+1})$  by Lemma 5,  $p'_i p'_{i+1} \notin E$  only in the case that any of  $p'_i$  and  $p'_{i+1}$  is a sharp vertex. By Lemma 6 this is impossible. So,  $p'_i p'_{i+1} \in E$ . This proves the first part of the lemma.

For all  $i \in [0, k]$ ,  $h(p'_i) \geq h(p'_{i+1})$  (Lemma 5), and neither  $p'_i$  nor  $p'_{i+1}$  is a sharp vertex (Lemma 6). Therefore, by Lemma 3, there exists a gently descending path (more precisely, an SGDP) from  $p'_i$  to  $p'_{i+1}$ . Clearly  $P'$  represents the gently descending path formed by concatenating all these “smaller” paths. □

**Lemma 8.** *Path  $P'$  represents a  $(1 + \epsilon)$ -SGDP.*

*Proof.* We first show that  $\sum_{i=1}^k \|p_i p'_i\| < \frac{\epsilon h}{2}$ . If  $p_i \neq p'_i$ , and  $e_i$  is a non-level edge, we have:  $|h(p_i) - h(p'_i)| \leq \delta$  by construction, and  $\frac{|h(p_i) - h(p'_i)|}{|p_i p'_i|} \geq \cos \theta$ , which implies that  $|p_i p'_i| \leq \delta \sec \theta$ . If  $p_i = p'_i$ , or  $e_i$  is a level edge,  $|p_i p'_i| \leq \delta \sec \theta$  in a trivial manner. Therefore,  $\sum_{i=1}^k |p_i p'_i| \leq k \delta \sec \theta$ . Since  $P$  is an ideal SGDP,  $k < 2n$  (the number of faces), and hence,  $\sum_{i=1}^k |p_i p'_i| < 2n \delta \sec \theta = \frac{\epsilon h \cos \psi}{2}$ . Observation 1(iii) implies  $\sum_{i=1}^k \|p_i p'_i\| \leq \sum_{i=1}^k (|p_i p'_i| \sec \psi) < \frac{\epsilon h \cos \psi}{2} \sec \psi = \frac{\epsilon h}{2}$ .

The length of  $P'$  is equal to:

$$\begin{aligned} \sum_{i=0}^k \|p'_i p'_{i+1}\| &\leq \sum_{i=0}^k (\|p'_i p_i\| + \|p_i p_{i+1}\| + \|p_{i+1} p'_{i+1}\|) \quad (\text{Observation 1(ii)}) \\ &= \sum_{i=0}^k \|p_i p_{i+1}\| + 2 \sum_{i=1}^k \|p_i p'_i\| \quad (\text{Observation 1(i)}) \\ &< \sum_{i=0}^k \|p_i p_{i+1}\| + \epsilon h . \end{aligned}$$

Assuming that  $P$  crosses at least one edge of the terrain (otherwise, both  $P'$  and  $P$  will have length  $\|sv\|$ ),  $h \leq \sum_{i=0}^k |p_i p_{i+1}| \leq \sum_{i=0}^k \|p_i p_{i+1}\|$  (Observation 1(iii)), and therefore,  $\sum_{i=0}^k \|p'_i p'_{i+1}\| < (1 + \epsilon) \sum_{i=0}^k \|p_i p_{i+1}\|$ . The proof then follows from Lemma 7.  $\square$

**Lemma 9.** *Let  $X = \frac{L}{h} \sec \theta \sec \psi$ . Graph  $G$  has  $O\left(\frac{n^2 X}{\epsilon}\right)$  nodes in total, and  $O\left(\frac{nX}{\epsilon}\right)$  nodes along any edge of the terrain.*

*Proof.* (Idea) The proof is the same as that of Lemma 7 in Ahmed et al. [1], except that we use  $\delta$  and  $X$  defined here.  $\square$

**Theorem 1.** *Let  $X = \frac{L}{h} \sec \theta \sec \psi$ . Given a vertex  $s$ , we can preprocess the terrain in  $O\left(\frac{n^2 X}{\epsilon} \log\left(\frac{nX}{\epsilon}\right)\right)$  time after which we can determine a  $(1 + \epsilon)$ -approximate SGDP from  $s$  to any query point  $v$  in: (i)  $O(nd)$  time if  $v$  is a vertex or a Steiner point, and (ii)  $O\left(n\left(d + \frac{X}{\epsilon}\right)\right)$  time otherwise.*

*Proof.* We first show that the path  $P''$  returned by our algorithm represents a  $(1 + \epsilon)$ -approximation of  $P$ . Path  $P''$  represents a gently descending path because any path in  $G$  represents a gently descending path in the terrain, and also the last segment of  $P''$  represents a gently descending path. It follows from the construction of  $P''$  that the length of  $P''$  is at most that of  $P'$ , and hence by Lemma 8,  $P''$  is a  $(1 + \epsilon)$ -approximation of  $P$ .

The preprocessing time of our algorithm is the same as the running time of the Bushwhack algorithm, which is  $O(|V| \log |V|) = O\left(\frac{n^2 X}{\epsilon} \log\left(\frac{nX}{\epsilon}\right)\right)$  by Lemma 9.

During the query phase, if  $v$  is a vertex or a Steiner point, the approximate path is in the tree  $T$ . Because the tree has height  $O(n)$ , it takes  $O(n)$  time to trace the path in the tree. Tracing the corresponding path in the terrain takes  $O(nd)$  time by Lemma 3. The total query time is thus  $O(nd)$  in this case. If  $v$  is neither a vertex nor a Steiner point,  $v$  is an interior point of a face or an edge of the terrain. The last intermediate node  $u$  on the path to  $v$  is a vertex or a Steiner point that lies on the boundary of a face containing  $v$ . If  $v$  is interior to a face [an edge], there are 3 [respectively 4] edges of the terrain on which  $u$  can lie. Thus there are  $O\left(\frac{nX}{\epsilon}\right)$  choices for  $u$  by Lemma 9, and we try all of them to find the shortest approximate distance from  $s$  to  $v$ . Finally tracing the corresponding path in the terrain takes  $O(nd)$  time by Lemma 3. The total query time in this case is  $O\left(\frac{nX}{\epsilon}\right) + O(nd) = O\left(n\left(d + \frac{X}{\epsilon}\right)\right)$ .  $\square$

**Corollary 1.** *If the answer to a query is the length of an SGDP (rather than the SGDP itself), the query times for Cases (i) and (ii) of Theorem 1 become  $O(1)$  and  $O\left(\frac{nX}{\epsilon}\right)$  respectively.*

## 6 Approximation using Non-uniform Steiner Points

Our second approximation algorithm differs from the first one only in the way Steiner points are placed. We now place Steiner points in two phases. First, on every edge  $e = v_1v_2$  we place Steiner points at points  $p \in e$  such that  $|pq| = \delta_1(1 + \delta_2)^i$  for  $q \in \{v_1, v_2\}$  and  $i \in \{0, 1, 2, \dots\}$ , where  $\delta_1 = \frac{\epsilon h}{6n} \cos \psi$  and  $\delta_2 = \frac{\epsilon h}{6L} \cos \psi$ . In the second phase we slice the terrain with a level plane through every Phase 1 Steiner point and every vertex, and add Steiner points at the points where these planes intersect the terrain.

**Theorem 2.** *Let  $X' = \frac{L}{h} \sec \psi$ . Given a vertex  $s$ , we can preprocess the terrain in  $O\left(\frac{n^2 X'}{\epsilon} \log^2\left(\frac{n X'}{\epsilon}\right)\right)$  time after which we can determine a  $(1 + \epsilon)$ -approximate SGDP from  $s$  to any point  $v$  in: (i)  $O(nd)$  time if  $v$  is a vertex or a Steiner point, and (ii)  $O\left(nd + \frac{n X'}{\epsilon} \log\left(\frac{n X'}{\epsilon}\right)\right)$  time otherwise.*

*Proof.* (Idea) The proof is similar to that of Theorem 1, but we use slightly different versions of Lemmas 8 and 9, as is done in the proof of Theorem 2 in Ahmed et al. [1].  $\square$

**Corollary 2.** *If the answer to a query is the length of an SGDP, the query times for Cases (i) and (ii) of Theorem 2 become  $O(1)$  and  $O\left(\frac{n X'}{\epsilon} \log\left(\frac{n X'}{\epsilon}\right)\right)$  respectively.*

## References

1. Mustaq Ahmed, Sandip Das, Sachin Lodha, Anna Lubiw, Anil Maheshwari, and Sasanka Roy. Approximation algorithms for shortest descending paths in terrains. *CoRR*, 0805.1401v1 [cs.CG], May 2008.

2. Mustaq Ahmed and Anna Lubiw. Shortest descending paths through given faces. In *Proceedings of the 18th Canadian Conference on Computational Geometry*, pages 35–38, August 2006. Accepted for publication in CCCG '06 Special Issue of *Computational Geometry: Theory and Applications*.
3. Mustaq Ahmed and Anna Lubiw. An approximation algorithm for shortest descending paths. *CoRR*, 0705.1364v1 [cs.CG], May 2007.
4. Mustaq Ahmed and Anna Lubiw. Properties of shortest descending paths. The 17th Fall Workshop on Computational and Combinatorial Geometry, Hawthorne, New York, November 2007. Extended abstract.
5. Mustaq Ahmed and Anna Lubiw. Shortest anisotropic paths with few bends is NP-complete. The 18th Fall Workshop on Computational Geometry, Troy, New York, October 2008. Extended abstract, submitted.
6. Lyudmil Aleksandrov, Anil Maheshwari, and Jörg-Rüdiger Sack. Determining approximate shortest paths on weighted polyhedral surfaces. *J. ACM*, 52(1):25–53, 2005.
7. Siu-Wing Cheng, Hyeon-Suk Na, Antoine Vigneron, and Yajun Wang. Approximate shortest paths in anisotropic regions. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 766–774, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.
8. Mark de Berg and Marc J. van Kreveld. Trekking in the Alps without freezing or getting tired. *Algorithmica*, 18(3):306–323, 1997.
9. Mark de Berg, Marc J. van Kreveld, Mark Overmars, and Otfried Cheong. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Berlin, Germany, 2nd edition, 2000.
10. Mark Lanthier, Anil Maheshwari, and Jörg-Rüdiger Sack. Shortest anisotropic paths on terrains. In *Proceedings of the 26th International Colloquium on Automata, Languages and Programming*, pages 524–533, London, UK, 1999. Springer-Verlag.
11. Joseph S. B. Mitchell and Christos H. Papadimitriou. The weighted region problem: finding shortest paths through a weighted planar subdivision. *J. ACM*, 38(1):18–73, 1991.
12. Neil C. Rowe and Ron S. Ross. Optimal grid-free path planning across arbitrarily-contoured terrain with anisotropic friction and gravity effects. *IEEE Trans. Robot. Autom.*, 6(5):540–553, 1990.
13. Sasanka Roy, Sandip Das, and Subhas C. Nandy. Shortest monotone descent path problem in polyhedral terrain. *Comput. Geom. Theory Appl.*, 37(2):115–133, 2007.
14. Sasanka Roy, Sachin Lodha, Sandip Das, and Anil Maheshwari. Approximate shortest descent path on a terrain. In *Proceedings of the 19th Canadian Conference on Computational Geometry*, pages 189–192, August 2007.
15. Zheng Sun and Tian-Ming Bu. On discretization methods for approximating optimal paths in regions with direction-dependent costs. *Inform. Process. Lett.*, 97(4):146–152, 2006.
16. Zheng Sun and John H. Reif. BUSHWHACK: An approximation algorithm for minimal paths through pseudo-Euclidean spaces. In *Proceedings of the 12th International Symposium on Algorithms and Computation*, pages 160–171, London, UK, 2001. Springer-Verlag.
17. Zheng Sun and John H. Reif. On finding energy-minimizing paths on terrains. *IEEE Transactions on Robotics*, 21(1):102–114, 2005.
18. Zheng Sun and John H. Reif. On finding approximate optimal paths in weighted regions. *J. Algorithms*, 58(1):1–32, 2006.