Area-Preserving Approximations of Polygonal Paths *

Prosenjit Bose[1] Sergio Cabello[2] Otfried Cheong[3] Joachim Gudmundsson[4]
Marc van Kreveld[5] Bettina Speckmann[4]

[1] School of Computer Science, Carleton University, Canada.
jit@scs.carleton.ca

[2] Department of Mathematics, Institute for Mathematics, Physics and Mechanics, Slovenia.
sergio.cabello@imfm.uni-lj.si

[3] Division of Computer Science, KAIST, Korea.
otfried@tclab.kaist.ac.kr

[4] Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands.
h.j.gudmundsson@tue.nl speckman@win.tue.nl

[5] Institute for Information and Computing Sciences, Utrecht University, The Netherlands.
marc@cs.uu.nl

Abstract

Let P be an x-monotone polygonal path in the plane. For a path Q that approximates P let $W_A(Q)$ be the area above P and below Q, and let $W_B(Q)$ be the area above Q and below P.

Given P and an integer k, we show how to compute a path Q with at most k edges that minimizes $W_A(Q)+W_B(Q)$. Given P and a cost C, we show how to find a path Q with the smallest possible number of edges such that $W_A(Q)+W_B(Q) \leq C$.

However, given P, an integer k, and a cost C, it is NP-hard to determine if a path Q with at most k edges exists such that $\max\{W_A(Q), W_B(Q)\} \leq C$. We describe an approximation algorithm for this setting.

Finally, it is also NP-hard to decide whether a path Q exists such that $|W_A(Q) - W_B(Q)| = 0$. Nevertheless, in this error measure we provide an algorithm for computing an optimal approximation up to an additive error.

1 Introduction

Let P be an x-monotone polygonal path. We consider the problem of approximating P by a "simpler" polygonal path Q. Imai and Iri [11, 12] introduced two different versions of this problem. In the first one, one is given an integer k and the aim is to compute a polygonal path Q that has k edges and approximates P in the best possible way according to some measure that compares P and Q. In the second version, one is given a tolerance $\varepsilon > 0$ and wants to compute a polygonal path Q that approximates P within ε and has the fewest vertices. Both versions have been considered for different measures that are based on variations of the notion of minimum distance between P and Q [1, 2, 3, 4, 8, 9, 10, 11, 12, 15, 16]. These problems have many applications in map simplification.

In this paper, we consider *area-preserving approximations* of polygonal paths. Area-preserving simplifications are particularly meaningful for a path representing the border between two countries or regions; the approximations simplify such paths without substantially distorting the area information. In quantitative mapping [6], for instance, preservation of area is more important than shape fidelity. Although area as a measure for line simplification has been proposed before [5, 14], the problem has not been studied algorithmically yet. We provide the first efficient algorithms with performance guarantees on the total area displaced by the simplification.

*J.G. supported by the Netherlands Organisation for Scientific Research (NWO), S.C. partially supported by the European Community Sixth Framework Programme under a Marie Curie Intra-European Fellowship.

We have two restrictions in our model for line simplification, both of which are quite common: The original path should be x-monotone and we do not allow Steiner points in the simplification. The first assumption is made to avoid that the simplified path has self-intersections, which leads to topological inconsistency in the output. To avoid this issue, most authors that consider distance error assume that the input is an x-monotone path; certain versions of optimal simplification without self-intersections are even NP-hard [7]. The second assumption allows us to reduce the amount of freedom that the simplification has and so enables more efficient approaches to the problem.

Problem formulation. Let P be an x-monotone polygonal path with vertices $p_1, \dots, p_n$ and let $Q = (p_{i_0}, \dots, p_{i_k})$ be an approximating path with $1 = i_0 < \cdots < i_k = n$. Let $W_A(Q)$ be the area above P and below Q, and let $W_B(Q)$ be the area above Q and below P. We consider three cost functions to measure the quality of the approximation Q.

Sum-area model. In this model the error of the approximation path Q is $W_A(Q) + W_B(Q)$. In other words, we wish to minimize the total transfer of area between the regions above and below P. We show that optimal approximations in this model can be computed in polynomial time. More precisely, given P and k, we can compute the minimum-error path Q with at most k edges in time $O(kn^2 + n^{2+\varepsilon})$, for any $\varepsilon > 0$. Furthermore, given P and a cost $C > 0$, we can compute the path Q with error at most C with the minimum number of links within the same time bound (here k is the number of edges in the solution). The sum-area model is discussed in Section 2.

Max-area model. If the regions above and below P are countries and Q is intended as a border simplification, then each country may feel badly only about the area it gives away. Here we model this situation: The error of the approximation path Q is $\max\{W_A(Q), W_B(Q)\}$. We show that it is NP-hard to compute the minimum-error k-link approximation path in this model. However, we also show that given P, k, and $\delta > 0$ one can compute an approximation path Q whose error is at most a factor $1 + \delta$ from the optimum error in $O(k^2n^2/\delta + n^{2+\varepsilon})$ time. The max-area model is discussed in Section 3.

Diff-area model. Another error measure that suggests itself is the cost $|W_A(Q) - W_B(Q)|$ which measures by how much the area above (or below) the path is changed by the simplification. This differs from the models above in that area is now "exchangeable", that is, area displaced at one spot can be compensated for at another spot. We show that it is NP-hard to decide whether a path Q exists such that $|W_A(Q) - W_B(Q)| = 0$. Nevertheless, given a polygonal path P whose convex hull has area H, k, and $\delta > 0$ we can compute an approximation path Q whose error is at most $\delta \cdot H$ larger than the optimal error in $O(k^2n^2/\delta)$ time. The diff-area model is discussed in Section 4.

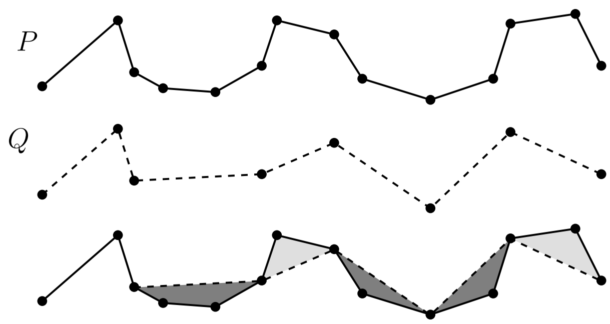


Figure 1: A path P and an approximation Q; $W_A(Q)$— dark grey, $W_B(Q)$—light grey.

2 The sum-area model

The following theorem constitutes the main result of this section.

Theorem 1 *Given a polygonal path P and an integer k, the optimal approximating path with at most k edges in the sum-area model can be computed in $O(kn^2 + n^{2+\varepsilon})$ time using $O(n^{2+\varepsilon})$ space, for any $\varepsilon > 0$.*

The algorithm consists of two steps. In the first step we construct a directed, weighted graph G_P from P. Its vertices are the vertices of P and it includes a directed edge $e = (p_i, p_j)$ for each pair $1 \le i < j \le n$. Each edge e of G_P has two weights, $w_a(e)$ and $w_b(e)$, which correspond to the area above P and below e, and vice versa (see Fig. 2(a)).

Any path in G_P from p_1 to p_n corresponds to a simplified path Q and we have $W_A(Q) = \sum_{e \in Q} w_a(e)$ and $W_B(Q) = \sum_{e \in Q} w_b(e)$. We can therefore solve our optimization problem in the second step by computing a shortest k-link path from p_1 to p_n in G_P, where the cost of an edge e is $w_a(e) + w_b(e)$.

Since our approximation path Q will only make use of edges from the graph G_P, we can easily accommodate additional constraints by restricting the set of edges of G_P. For instance, if we require the path Q to stay within a certain Hausdorff-distance of P, or if certain landmark features may not be displaced, this can be handled by removing the edges of G_P that violate these constraints.

2.1 Constructing G_P

The weights w_a (and symmetrically w_b) of G_P are computed as follows:

1. Let R be the polygon formed with the polyline P, two vertical segments at p_1 and p_n, and a horizontal segment above P and connecting the two vertical segments (see Fig. 2(b)).

2. Recursively partition R by a vertical straight line segment into two polygons R_l and R_r of the same complexity. This partition gives a hierarchical decomposition, denoted T, into $O(n)$ polygons $R_1, \ldots, R_\ell$ with total complexity $O(n \log n)$.

3. The weight $w_a(e)$ of an edge $e = (p_i, p_j)$ can be expressed as the sum over $O(\log n)$ *halfplane area queries* of the type: given a polygon R_i and halfplane h, determine the area of $R_i \cap h$. Instead of answering each query on-line we save (batch) the queries to each subpolygon R_i. When all queries have been saved we use the following result by Langerman:

   Fact 1 (Langerman [13]) *Given a simple polygon P with n vertices and m lines, the area of P on both sides of each line can be computed in time $O(m^{2/3}n^{2/3+\varepsilon} + (n+m)$ polylog $n)$ for any $\varepsilon > 0$.*

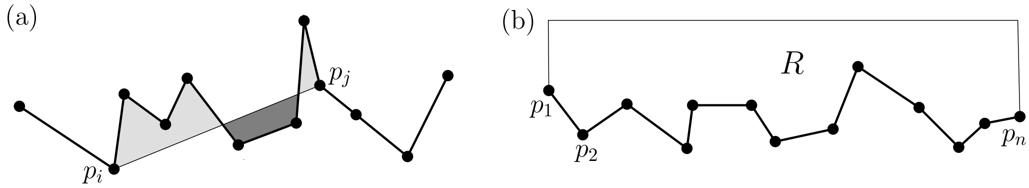


Figure 2: (a) The weights of $e = (p_i, p_j)$ in G_P: $w_a(e)$—dark gray, $w_b(e)$—light gray. (b) Constructing the polygon R from P.

Lemma 2 *The graph G_P can be constructed in $O(n^{2+\varepsilon})$ time and space.*

Proof. Recall that the tree T is a binary tree with $O(n)$ leaves and that the total complexity of all polygons in T is $O(n \log n)$. As mentioned above, each weight $w_a(e)$ can be expressed as the sum over $O(\log n)$ halfplane queries to some polygons R_i. Since there are $O(n^2)$ edges there will be $O(n^2 \log n)$ halfplane area queries to $O(n)$ polygons, i.e., $O(n^2 \log n)$ queries at the nodes of T.

The total time-complexity is obtained by summing all the queries using Fact 1. For a node v in T, let m_v be the number of queries batched on it. If $|R_v|$ denotes the complexity of the polygon R_v, then the total time needed is

$$O\Big(\sum_{v \in T} (m_v + |R_v|) \operatorname{polylog} n + m_v^{2/3} |R_v|^{2/3+\varepsilon}\Big).$$

Since $|R_v| = O(n)$ we have that $O\big(\sum_{v\in T}((m_v + |R_v|) \operatorname{polylog} n\big) = O(n^{2+\varepsilon})$. It remains to bound $\sum_{v\in T} m_v^{2/3} |R_v|^{2/3+\varepsilon}$. Consider a level $\mathcal{L}$ in the tree T and let h be the number of nodes in $\mathcal{L}$. For any node $v \in \mathcal{L}$ we have $|R_v| = O(n/h)$ and $\sum_{v\in\mathcal{L}} m_v = O(n^2)$. The cost at level $\mathcal{L}$ is therefore

$$O\Big(\sum_{v \in \mathcal{L}} m_v^{2/3} |R_v|^{2/3+\varepsilon}\Big) = O\Big((n/h)^{2/3+\varepsilon} \cdot \sum_{v\in\mathcal{L}} {m_v}^{2/3}\Big).$$

This value is maximized when all the m_v's have the same value, that is, when $m_v = O(n^2/h)$, so it follows that at level $\mathcal{L}$ the time needed is bounded by

$$O\Big((n/h)^{2/3+\varepsilon} \cdot \sum_{v\in\mathcal{L}} (n^2/h)^{2/3}\Big) = O(n^{2+\varepsilon}).$$

Since T has depth $O(\log n)$, we obtain the lemma. □

2.2 Computing the optimum k-link path

We use dynamic programming to find a minimum-cost path in G_P from p_1 to p_n consisting of at most k edges. Observe that once the last edge (p_i, p_n) of the simplification path Q has been fixed, the path from p_1 to p_i is a minimum-cost path of at most $k-1$ edges.

Let $L[i,t]$ be the cost of an optimal path from p_1 to p_i with at most t edges, for $1 < i \le n$, $1 \le t \le k$. The values $L[i,t]$ are computed recursively as follows: If $t = 1$ then $L[i,1] = w_a(p_1, p_i) + w_b(p_1, p_i)$. If $t > 1$ then

$$L[i,t] = \min\big\{L[i,t-1], \min_{1<j<i}\{L[j,t-1] + w_a(p_j,p_i) + w_b(p_j,p_i)\}\big\}.$$

There are $O(kn)$ L-values to compute and each computation takes $O(n)$ time. Theorem 1 follows.

2.3 Minimizing the number of edges

We can use the same approach to minimize the number of edges of Q for a given approximation error C. We compute $L[\cdot,k]$ for increasing values of k until we find a k such that $L[n,k] \le C$. Since G_P needs to be computed only once, this proves the following result.

Corollary 3 *Given a polygonal path P and a cost C, an approximating path in the sum-area model with cost at most C and the minimum number of edges can be computed in $O(kn^2+n^{2+\varepsilon})$ time using $O(n^{2+\varepsilon})$ space, for any $\varepsilon > 0$, where k is the number of edges in the solution.*

3 The max-area model

We first show that area-preserving path approximation is NP-hard in the max-area model and then describe an approximation algorithm.

3.1 NP-hardness

Recall that the following PARTITION-problem is (weakly) NP-hard: Given a set of natural numbers $A = \{a_1, \ldots, a_n\}$, partition it into two disjoint sets A_1 and A_2 such that $A = A_1 \cup A_2$ and $\sum_{a_i \in A_1} a_i = \sum_{a_i \in A_2} a_i$.

Theorem 4 *Given a polygonal path P, an integer k, and a cost C, it is NP-hard to determine whether a simplification Q with at most k edges and approximation error at most C exists in the max-area model.*

Proof. The proof is a reduction from PARTITION. Given an instance of PARTITION, let $S := \sum_{a_i \in A} a_i$ and set $K := 2S + 1$. For each a_i let P_i be the path from p_i to p_{i+1} sketched in Fig. 3. The precise coordinates of the points are given there as well. Let P be the concatenation of the pieces $P_1, \ldots, P_n$. Each P_i has six edges, hence P consists of $6n$ edges. Note that the two triangles $\triangle p_i q_i r_i$ and $\triangle q_i r_i s_i$ both have area a_i (see Fig. 3 right).

Consider now a simplification Q of P with max-area error at most $S/2$. Then Q cannot contain any shortcut in the subpath P_i, except possibly between p_i and r_i, or between q_i and s_i; also, Q cannot contain a shortcut between a point in P_i and a point in P_j, for $i \neq j$. Indeed, any simplification with such a shortcut e would have $\max\{w_a(e), w_b(e)\} \geq K/4$ and therefore $\max\{W_A(Q), W_B(Q)\} \geq K/4 > S/2$.

It follows that Q must go through all the points p_i, $1 \leq i \leq n+1$. Let Q_i be the subpath of Q from p_i to p_{i+1}. Each Q_i consists of at least 5 edges, and so Q has at least $5n$ edges.

If Q has exactly $5n$ edges, it follows that each Q_i has exactly 5 edges, and is either of the form $p_i r_i s_i \cdots p_{i+1}$, or of the form $p_i q_i s_i \cdots p_{i+1}$. In the first case, the approximation error of Q_i is $W_A(Q_i) = 0$, $W_B(Q_i) = a_i$, in the second case, we have $W_A(Q_i) = a_i$, $W_B(Q_i) = 0$. Since $\sum_{i=1}^{n} a_i = S$, we can have $\max\{W_A(Q), W_B(Q)\} \leq S/2$ if and only if A can be split into two disjoint subsets A_1 and A_2 such that $\sum_{a_i \in A_1} a_i = \sum_{a_i \in A_2} a_i = S/2$. $\square$

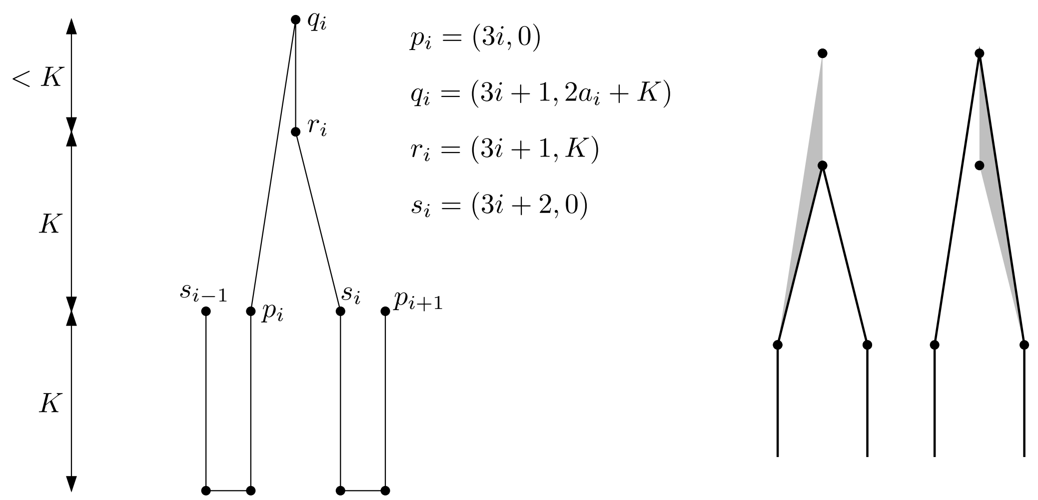


Figure 3: Sketch of the reduction for the NP-hardness proof in the max-area model. Only the shortcuts shown on the right can be used because all others move too much area from above to below or vice versa.

3.2 An approximation algorithm

We will make use of *rounding* to find an approximate solution. We are given P, an integer k, and a parameter δ, and our goal is to find a path Q with at most k edges such that $\max\{W_A(Q), W_B(Q)\} \leq (1+\delta)\cdot T$, where $T := \min_Q \max\{W_A(Q), W_B(Q)\}$ is the value of the optimal solution.

We start by running the algorithm of Theorem 1, which constructs the graph G_P and computes the value $M := \min_Q\{W_A(Q) + W_B(Q)\}$. Clearly we have $M/2 \leq T \leq M$.

Let now $\Delta := \delta M/2k$. For each edge e of G_P, we compute values $\overline{w_a}(e)$ and $\overline{w_b}(e)$ as follows: if $w_a(e) > M$, then $\overline{w_a}(e) = \infty$, otherwise $\overline{w_a}(e) := \lfloor w_a(e)/\Delta\rfloor$. Similarly $\overline{w_b}(e) = \infty$ if $w_b(e) > M$, otherwise $\overline{w_b}(e) := \lfloor w_b(e)/\Delta\rfloor$.

We observe that all finite values of $\overline{w_a}(e)$ and $\overline{w_b}(e)$ are integers in the range $0, 1, \ldots, \lfloor M/\Delta\rfloor$. This implies that one can avoid to actually use the floor function by replacing it with binary search. Each floor computation requires $\log(M/\Delta) = \log(2k/\delta) = O(\log k + \log(1/\delta))$ search steps, for a total of $O(n^2 \log k + n^2 \log(1/\delta))$ additional time.

Let now Q be a path in G_P with $\max\{W_A(Q), W_B(Q)\} \leq M$ and with at most k edges. Let $\overline{W_A}(Q) := \sum_{e\in Q} \overline{w_a}(e)$, and $\overline{W_B}(Q) := \sum_{e\in Q} \overline{w_b}(e)$. Note that for any $e \in Q$, $\overline{w_a}(e)$ is finite, and we have

$$\Delta\overline{w_a}(e) \leq w_a(e) < \Delta\overline{w_a}(e) + \Delta.$$

This implies

$$\Delta\overline{W_A}(Q) = \Delta\sum_{e\in Q}\overline{w_a}(e) \leq \sum_{e\in Q} w_a(e) = W_A(Q),$$

and

$$\begin{aligned} W_A(Q) &= \sum_{e\in Q} w_a(e) < \sum_{e\in Q}\big(\Delta\overline{w_a}(e) + \Delta\big) \leq \Delta\sum_{e\in Q}\overline{w_a}(e) + k\Delta = \Delta\overline{W_A}(Q) + k\Delta \\ &= \Delta\overline{W_A}(Q) + k\delta M/2k = \Delta\overline{W_A}(Q) + \delta M/2 \leq \Delta\overline{W_A}(Q) + \delta T, \end{aligned}$$

(using $M/2 \leq T$ in the last step). Similarly, we conclude that

$$\Delta\overline{W_B}(Q) \leq W_B(Q) < \Delta\overline{W_B}(Q) + \delta T.$$

Let now Q_{app} be a path minimizing $T^* := \max\{\overline{W_A}(Q_{\text{app}}), \overline{W_B}(Q_{\text{app}})\}$ with at most k edges. We claim that Q_{app} is the approximate solution we are looking for. Indeed, let Q_{opt} be the optimal solution. Then $\Delta\overline{W_A}(Q_{\text{opt}}) \leq W_A(Q_{\text{opt}}) \leq T$, and $\Delta\overline{W_B}(Q_{\text{opt}}) \leq W_B(Q_{\text{opt}}) \leq T$, and therefore $\Delta T^* \leq T$. This implies

$$W_A(Q_{\text{app}}) \leq \Delta\overline{W_A}(Q_{\text{app}}) + \delta T \leq \Delta T^* + \delta T \leq T + \delta T = (1+\delta)T.$$

Similarly, $W_B(Q_{\text{app}}) \leq (1+\delta)T$, which proves the claim.

It remains to show how to compute the path Q_{app}. Again, we employ dynamic programming. Let $F := \lfloor M/\Delta\rfloor$ and recall that all finite $\overline{w_a}(e)$, $\overline{w_b}(e)$ are integers in the range $0, \ldots, F$. For integers i, t, s with $1 < i \leq n$, $1 \leq t \leq k$, and $0 \leq s \leq F$, let $L[i,t,s]$ be the minimum value of $\overline{W_B}(Q)$ for any path Q from p_1 to p_i with at most t edges and the restriction $\overline{W_A}(Q) \leq s$. We compute $L[i,t,s]$ recursively as follows: if $t = 1$, then $L[i,1,s] = \overline{w_b}(p_1,p_i)$ if $\overline{w_a}(p_1,p_i) \leq s$, otherwise $L[i,1,s] = \infty$. If $t > 1$, then

$$L[i,t,s] = \min\Big\{L[i,t-1,s], \min_{\substack{1<j<i \\ \overline{w_a}(p_j,p_i)\leq s}} \{L[j,t-1,s-\overline{w_a}(p_j,p_i)] + \overline{w_b}(p_j,p_i)\}\Big\}.$$

There are $O(nkF) = O(nk^2/\delta)$ L-values, each of which can be computed in time $O(n)$, for a total running time of $O(n^2k^2/\delta)$. Once we have computed all values, we can determine the optimal cost T^* using the relation

$$T^* = \min_{0 \le s \le F} \max\{s, L[n, k, s]\}.$$

Observe that for computing $L[\cdot, t, \cdot]$ we only use the values $L[\cdot, t-1, \cdot]$. This means that during the computation of T^* we need to store $O(nk/\delta)$ values only. To reconstruct the path Q_{app}, however, we do need to store the entire table $L[\cdot, \cdot, \cdot]$. We conclude with the following theorem.

Theorem 5 *Given a polygonal path P, an integer k, and a parameter $\delta > 0$, an approximating path with at most k edges and with max-area cost at most $1 + \delta$ times the optimal can be computed in $O(k^2n^2/\delta + n^{2+\varepsilon})$ time using $O(n^{2+\varepsilon} + nk^2/\delta)$ space, for any $\varepsilon > 0$.*

4 The diff-area model

We first show that area-preserving path approximation is NP-hard in the diff-area model and then describe an algorithm which computes an approximation within an additive error of the optimal one.

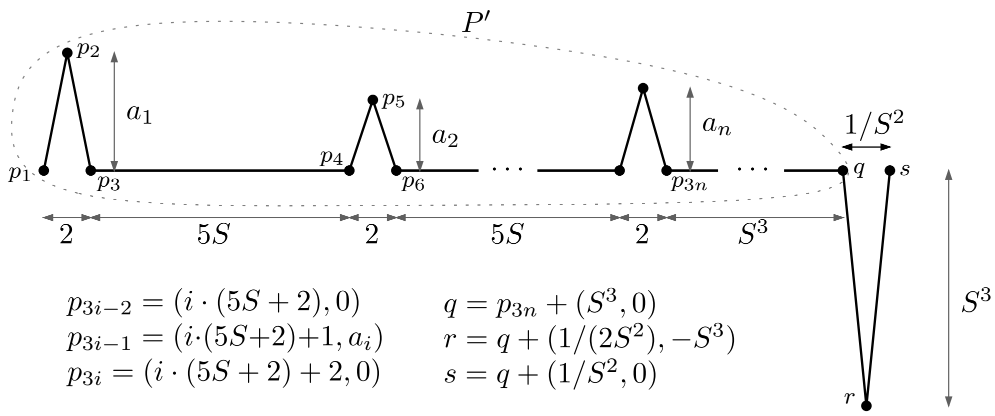
4.1 NP-hardness

Theorem 6 *Given a polygonal path P, it is NP-hard to determine whether an approximation Q with fewer edges and $W_A(Q) - W_B(Q) = 0$ exists.*

Proof. The proof is again a reduction from PARTITION. Given an instance $A = \{a_1, \ldots, a_n\}$ of PARTITION where each a_i is a positive natural number, let $S := \sum a_i$ and consider the path P described in Figure 4. The exact coordinates of the vertices are depicted in the figure. Observe that for $e = (p_{3i-2}, p_{3i})$ we have $w_a(e) = 0, w_b(e) = a_i$. We also have $w_a(q, s) = S/2$ and $w_b(q, s) = 0$.

It is clear that if there is a subset $A_1 \subset A$ with $S/2 = \sum_{a_i \in A} a_i$, then the approximation Q that uses the shortcuts (p_{3i-2}, p_{3i}), for all $a_i \in A_1$, and the shortcut (q, s) has $W_A(Q) = W_B(Q)$. The remainder of the proof is devoted to show that (essentially) this is the only approximation Q with $W_A(Q) = W_B(Q)$.

We first make some observations regarding possible approximations of the initial part of P.



$p_{3i-2} = (i \cdot (5S + 2), 0)$ $\quad q = p_{3n} + (S^3, 0)$
$p_{3i-1} = (i \cdot (5S+2)+1, a_i)$ $\quad r = q + (1/(2S^2), -S^3)$
$p_{3i} = (i \cdot (5S + 2) + 2, 0)$ $\quad s = q + (1/S^2, 0)$

Figure 4: Sketch of the reduction for the NP-hardness proof in the diff-area model.

Claim 1 *Let p' be a vertex of P different from r or s, and let P' be the subpath of P from p_1 to p'. For any approximation Q' of P', we have*

(1) $W_B(Q') \leq S$,

(2) $W_A(Q') - W_B(Q') < 8S^4$, and

(3) if $W_A(Q') > 0$, then $W_A(Q') - W_B(Q') > S$.

The maximum possible value of $W_B(Q')$ is the area below P' but above the convex hull of P'. This is the total area of all the peaks, proving (1). To show (2), observe that the convex hull of P' has area bounded by $(n \cdot (5S+2) + S^3) \cdot \max_i\{a_i\} < (S \cdot (5S+2) + S^3) \cdot S < 8S^4$. Finally, if Q' has a shortcut $e \in Q'$ with $w_a(e) \neq 0$, then $w_a(e) > 2S$, and hence $W_A(Q') > 2S$. Since $W_B(Q') \leq S$ by (1), we conclude that $W_A(Q') - W_B(Q') > 2S - S = S$. This proves Claim 1.

Assume now that an approximation Q of P with $W_A(Q) - W_B(Q) = 0$ exists. We will show that the answer to the partition instance is *yes*. We distinguish two cases, depending on whether $r \in Q$ or $r \notin Q$.

If $r \in Q$, let p' be the vertex in Q before r, and let Q' be the subpath of Q from p_1 to p'. If $p' = q$, then $0 = W_A(Q) - W_B(Q) = W_A(Q') - W_B(Q')$. Since $Q \neq P$, this implies $W_A(Q) > 0$, and so by Claim 1 (3) $W_A(Q') - W_B(Q') > S > 0$, a contradiction. If $p' \neq q$, then note that $0 = W_A(Q) - W_B(Q) = W_A(Q') - W_B(Q') + w_a(p', r) - w_b(p', r)$. Since $w_b(p', r) > S^6/2$, $w_a(p', r) < 8S^4$ and Claim 1 (2), this implies $8S^4 > W_A(Q') - W_B(Q') = w_b(p', r) - w_a(p', r) > S^6/2 - 8S^4 > 8S^4$, a contradiction for $S > 5$.

If $r \notin Q$, let $\tilde{p}$ be the vertex in Q before s, let R be the subpath of Q from p_1 to $\tilde{p}$ concatenated with q, s, and let R' be the subpath of R from p_1 to q. We observe that $0 = W_A(Q) - W_B(Q) = W_A(R) - W_B(R) + \mathit{area}(\triangle \tilde{p}qs) = W_A(R') + S/2 - W_B(R') + \mathit{area}(\triangle \tilde{p}qs)$.

If the y-coordinate of $\tilde{p}$ is non-zero (and therefore positive), then $W_A(R') > 0$. By Claim 1 (3) with $p' = q$, we have $0 = W_A(R') + S/2 - W_B(R') + \mathit{area}(\triangle \tilde{p}qs) > S + S/2 > 0$, a contradiction.

Finally, if the y-coordinate of $\tilde{p}$ is zero, then $\mathit{area}(\triangle \tilde{p}qs) = 0$, and so $W_A(R') - W_B(R') = -S/2 < 0$. By Claim 1 (3) with $p' = q$, this is impossible if $W_A(R') > 0$, and so $W_A(R') = 0$, implying $W_B(R') = S/2$. This means that there is a subset $A_1 \subset A$ with $\sum_{a_i \in A_1} a_i = S/2$, and the answer to the partition problem is indeed *yes*. □

Since it is NP-hard to decide if the optimum is 0, we cannot approximate P within a multiplicative factor of the optimum. That is, for any function $f(n)$ it is NP-hard to compute an $f(n)$-approximation of P in the diff-area model.

4.2 A bounded-additive-error algorithm

We describe an algorithm that constructs an approximation of P within an additive error with respect to the optimal approximation. We follow the blueprint of Section 3.2.

For an edge $e = (p_i, p_j)$ of the graph G_P (see Section 2), let $w(e) = w_a(e) - w_b(e)$. We first show how the weights $w(e)$ can be computed efficiently, without the need to compute w_a and w_b as in Section 2.1.

Consider a polygon R above P as in Section 2.1. Let q_i be the intersection of the horizontal segment at the top of R with the vertical line through p_i; see Fig. 5. Let $R_{i,j}$ be the area of the polygon described by $p_i, p_{i+1}, \ldots, p_j, q_j, q_i, p_i$, and let $T_{i,j}$ be the area of the trapezoid described by p_i, p_j, q_j, q_i. We observe that $T_{i,j} = R_{i,j} + w_b(p_i, p_j) - w_a(p_i, p_j)$, and so

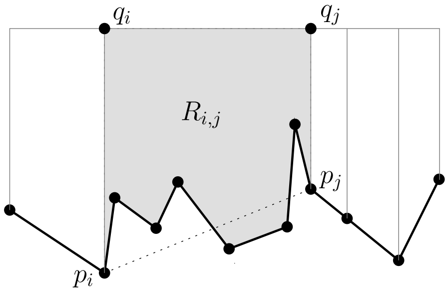


Figure 5: Computing $w(p_i, p_j)$.

$w(p_i, p_j) = R_{i,j} - T_{i,j} = R_{1,j} - R_{1,i} - T_{i,j}$. It follows that after precomputing and storing the values $R_{1,i}$, for $1 < i \le n$, we can return the weight $w(e)$ for any edge in constant time. The computation can be done in linear time in a single scan of $p_1, \ldots, p_n$.

We are given P, an integer k, and a parameter $\delta > 0$, and our goal is to find a path Q_{app} with at most k edges such that $|W_A(Q_{\text{app}}) - W_B(Q_{\text{app}})| \le \min_Q\{|W_A(Q) - W_B(Q)|\} + \delta H$, where H is the area of the convex hull of P.

Let $\Delta := \delta H/2k$. For an edge e of G_P, let $\overline{w}(e) := \lfloor w(e)/\Delta \rfloor$. Since $-H \le w(e) \le H$, the value of $\overline{w}(e)$ is an integer in the range $\lfloor -2k/\delta \rfloor, \ldots, \lfloor 2k/\delta \rfloor$. As in Section 3.2, we can avoid the use of the floor function by using binary search.

For a path Q, let $W(Q) := W_A(Q) - W_B(Q) = \sum_{e \in Q} w(e)$ and $\overline{W}(Q) := \sum_{e \in Q} \overline{w}(e)$. We have $\Delta\overline{w}(e) \le w(e) \le \Delta\overline{w}(e) + \Delta$, and summing over $e \in Q$ gives

$$\Delta\overline{W}(Q) \le W(Q) \le \sum_{e \in Q}(\Delta\overline{w}(e) + \Delta) = \Delta\overline{W}(Q) + k\Delta \le \Delta\overline{W}(Q) + \delta H/2.$$

This implies $|W(Q) - \Delta\overline{W}(Q)| \le \delta H/2$ for any approximating path Q, and therefore also $\big||W(Q)| - \Delta|\overline{W}(Q)|\big| \le \delta H/2$.

We will compute a path Q_{app} minimizing $|\overline{W}(Q_{\text{app}})|$. Let us first argue that this is the desired approximate solution. Indeed, let Q_{opt} be a path minimizing $|W(Q_{\text{opt}})|$, that is, a true optimal solution. We then have

$$|W(Q_{\text{app}})| \le \Delta|\overline{W}(Q_{\text{app}})| + \delta H/2 \le \Delta|\overline{W}(Q_{\text{opt}})| + \delta H/2 \le |W(Q_{\text{opt}})| + \delta H.$$

It remains to show how to compute Q_{app}. Once more, we employ dynamic programming. For integers i, t, s with $1 < i \le n$, $1 \le t \le k$, and $\lfloor -2k/\delta \rfloor \le s \le \lfloor 2k/\delta \rfloor$, let $L[i, t, s]$ be a boolean value that encodes if there is a path Q from p_1 to p_i with at most t edges and $\overline{W}(Q) = s$. We compute $L[i, t, s]$ recursively as follows: if $t = 1$, then $L[i, 1, s] = \textit{true}$ if $\overline{w}(p_1, p_i) = s$, and otherwise $L[i, 1, s] = \textit{false}$. If $t > 1$, then

$$L[i, t, s] := L[i, t-1, s] \vee \bigvee_{1<j<i} L[j, t-1, s - \overline{w}(p_j, p_i)].$$

There are $O(nk^2/\delta)$ L-values, each of which can be computed in time $O(n)$, for a total running time of $O(n^2k^2/\delta)$. Once we have computed all values, we can determine the approximately optimal cost $\overline{W}(Q_{\text{app}})$ using the relation

$$\overline{W}(Q_{\text{app}}) = \min\{|s| \mid L[n, k, s] = \textit{true}\}.$$

From the table $L[\cdot, \cdot, \cdot]$ it is easy to reconstruct the path Q_{app} itself. We conclude with the following theorem.

Theorem 7 *Given a polygonal path P whose convex hull has area H, an integer k, and a parameter $\delta > 0$, an approximating path with at most k edges and with diff-area cost at most δH larger than the optimal can be computed in $O(k^2n^2/\delta)$ time using $O(nk^2/\delta)$ space.*

5 Conclusions

We studied the complexity of polygonal path approximation under error measures that involve the displaced area. We discussed three models, provided a polynomial time algorithm for minimum link approximation in the first model, and showed NP-hardness for the other two models. For those models, we presented approximation algorithms. All algorithms are based on dynamic programming. Improving the efficiency of the algorithms is the main open problem. Another topic worth investigating is polygonal path approximation based on multiple criteria.

References

[1] H. Alt, J. Blömer, M. Godau, and H. Wagener. Approximation of convex polygons. In *Proceedings of the 17th International Colloquium on Automata, Languages and Programming*, volume 443 of *Lecture Notes in Computer Science*, pages 703–716, Berlin, 1990. Springer-Verlag.

[2] G. Barequet, M. T. Goodrich, D. Z. Chen, O. Daescu, and J. Snoeyink. Efficiently approximating polygonal paths in three and higher dimensions. In *Proceedings of the 14th Annual ACM Symposium on Computational Geometry*, pages 317–326, 1998.

[3] W. S. Chan and F. Chin. Approximation of polygonal curves with minimum number of line segments or minimum error. *Intern. Journal Comp. Geometry & Appl.*, 6:59–77, 1996.

[4] D. Z. Chen and O. Daescu. Space-efficient algorithms for approximating polygonal curves in two-dimensional space. *Intern. Journal Comp. Geometry & Appl.*, 13:95–111, 2003.

[5] R. G. Cromley. *Digital Cartography.* Prentice Hall, Englewood Cliffs, 1992.

[6] B. D. Dent. *Cartography - thematic map design.* McGraw-Hill, 5th edition, 1999.

[7] R. Estkowski and J. Mitchell. Symplifying a polygonal subdivision while keeping it simple. In *Proc. 17th Symp. Comp. Geometry*, pages 40–49, 2001.

[8] M. T. Goodrich. Efficient piecewise-linear function approximation using the uniform metric. *Discrete & Computational Geometry*, 14:445–462, 1995.

[9] J. Gudmundsson, G. Narasimhan, and M. Smid. Distance-preserving simplification of polygonal paths. In *Proc. 23rd Conf. Found. Software Techn. Theor. Comp. Science*, 2003.

[10] L. J. Guibas, J. E. Hershberger, J. S. B. Mitchell, and J. S. Snoeyink. Approximating polygons and subdivisions with minimum link paths. *Intern. Journal of Comp. Geometry & Appl.*, 3:383–415, 1993.

[11] H. Imai and M. Iri. Computational-geometric methods for polygonal approximations of a curve. *Computer Vision, Graphics and Image Processing*, 36:31–41, 1986.

[12] H. Imai and M. Iri. Polygonal approximations of a curve-formulations and algorithms. In G. T. Toussaint, editor, *Computational Morphology*, pages 71–86. North-Holland, Amsterdam, Netherlands, 1988.

[13] S. Langerman. The complexity of halfspace area queries. *Discrete & Computational Geometry*, 30:639–648, 2003.

[14] R. B. McMaster and K. S. Shea. *Generalization in Digital Cartography.* Association of American Cartographers, Washington D.C., 1992.

[15] A. Melkman and J. O'Rourke. On polygonal chain approximation. In *Computational Morphology*, pages 87–95. North-Holland, Amsterdam, 1988.

[16] K. R. Varadarajan. Approximating monotone polygonal curves using the uniform metric. In *Proceedings of the 12th Annual ACM Symposium on Computational Geometry*, pages 311–318, 1996.