Discrete Comput Geom (2011) 46:334–360
DOI 10.1007/s00454-011-9338-8

Representing a Functional Curve by Curves with Fewer Peaks

Danny Z. Chen · Chao Wang · Haitao Wang

Received: 27 March 2010 / Revised: 24 January 2011 / Accepted: 28 February 2011 /
Published online: 25 March 2011

Abstract In this paper, we study the problems of (approximately) representing a functional curve in 2-D by a set of curves with fewer peaks. Representing a function (or its curve) by certain classes of structurally simpler functions (or their curves) is a basic mathematical problem. Problems of this kind also find applications in applied areas such as intensity-modulated radiation therapy (IMRT). Let $\mathbf{f}$ be an input piecewise linear functional curve of size n. We consider several variations of the problems. (1) Uphill–downhill pair representation (UDPR): Find two nonnegative piecewise linear curves, one nondecreasing (uphill) and one nonincreasing (downhill), such that their sum exactly or approximately represents $\mathbf{f}$. (2) Unimodal representation (UR): Find a set of unimodal (single-peak) curves such that their sum exactly or approximately represents $\mathbf{f}$. (3) Fewer-peak representation (FPR): Find a piecewise linear curve with at most k peaks that exactly or approximately represents $\mathbf{f}$. Furthermore, for each problem, we consider two versions. For the UDPR problem, we study its *feasibility* version: Given $\epsilon > 0$, determine whether there is a feasible UDPR solution for $\mathbf{f}$ with an approximation error ϵ; its min-ϵ version: Compute the minimum approximation error ϵ^* such that there is a feasible UDPR solution for $\mathbf{f}$ with error ϵ^*. For the UR problem, we study its min-k version: Given $\epsilon > 0$, find a feasible solution

This research was supported in part by the National Science Foundation under Grants CCF-0515203 and CCF-0916606.
The work of H. Wang was also supported in part by a graduate fellowship from the Center for Applied Mathematics, University of Notre Dame.

D.Z. Chen · C. Wang · H. Wang (✉)
Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556, USA
e-mail: hwang6@nd.edu

D.Z. Chen
e-mail: dchen@nd.edu

C. Wang
e-mail: cwang1@nd.edu

with the minimum number k^* of unimodal curves for **f** with an error ϵ; its min-ϵ version: given $k > 0$, compute the minimum error ϵ^* such that there is a feasible solution with at most k unimodal curves for **f** with error ϵ^*. For the FPR problem, we study its min-k version: Given $\epsilon > 0$, find one feasible curve with the minimum number k^* of peaks for **f** with an error ϵ; its min-ϵ version: given $k \geq 0$, compute the minimum error ϵ^* such that there is a feasible curve with at most k peaks for **f** with error ϵ^*. Little work has been done previously on solving these functional curve representation problems. We solve all the problems (except the UR min-ϵ version) in optimal $O(n)$ time, and the UR min-ϵ version in $O(n + m \log m)$ time, where $m < n$ is the number of peaks of **f**. Our algorithms are based on new geometric observations and interesting techniques.

Keywords Curve approximation · Curve simplification · Curves with fewer peaks · Algorithm design

1 Introduction

1.1 Problem Descriptions

In this paper, we study the problems of exactly or approximately representing a 2-D functional curve by a set of curves with fewer peaks. Let **f** be an arbitrary input piecewise linear functional curve of size n. In general, when representing **f** by one or more structurally simpler curves, $\mathbf{g}^{(1)}, \mathbf{g}^{(2)}, \ldots, \mathbf{g}^{(k)}$ $(k \geq 1)$, we are interested in the following aspects of the representation: (1) the *representation mode*, which defines the types of and constraints on the simpler curves used, (2) the *representation complexity*, which is the number of simpler curves involved in the representation, and (3) the *representation error*, which is the vertical distance between the input functional curve **f** and the sum of the simpler curves in the representation, i.e., $\sum_{i=1}^{k} \mathbf{g}^{(i)}$.

For simplicity, we describe the input piecewise linear curve **f** by $(f_1, f_2, \ldots, f_n)$, where $f_i = \mathbf{f}(x_i)$ is the value of **f** at the ith x-coordinate x_i $(x_i < x_{i+1}$ for each $i)$. Without loss of generality (WLOG), the x_i's are all omitted in our discussion. For the consistency of our algorithmic manipulation and analysis, we need to define carefully the *peaks* of a functional curve $\mathbf{f} = (f_1, f_2, \ldots, f_n)$, with a little subtlety. Clearly, a peak is at a local maximal height. If multiple consecutive vertices of **f** all have the same local maximal height and if this group of vertices does not include the last vertex of **f**, then we define the *peak* for this group of vertices as only the *first* vertex of the group. However, if the group includes the last vertex of **f**, then we define the *peak* as the *last* vertex of the group (and of **f**). Figure 1 shows an example of peaks by our definition. The precise definition of peaks is as follows: we call f_i a *peak* of **f** if (1) $i = 1$ and there is a j with $1 < j \leq n$ such that $f_1 = \cdots = f_{j-1} > f_j$, or (2) $1 < i < n$, $f_{i-1} < f_i$ and there is a j with $i < j \leq n$ such that $f_i = \cdots = f_{j-1} > f_j$, or (3) $i = n$ and there is a j with $1 \leq j \leq n-1$ such that $f_j < f_{j+1} = \cdots = f_n$.

Specifically, we consider three modes of representation in this paper. (1) *Uphill–downhill pair representation* (*UDPR*): Represent a curve **f** by two curves, one nondecreasing (uphill) and one nonincreasing (downhill). (2) *Unimodal representation*

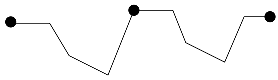

Fig. 1 Illustrating the peaks (the *black points*) on a piecewise linear curve

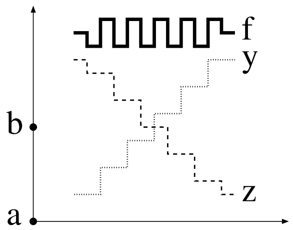

Fig. 2 An *uphill curve* **y** and a *downhill curve* **z** of size n each can sum up to form a functional curve **f** with $O(n)$ peaks

(*UR*): Represent **f** by a set of unimodal curves. A functional curve **g** is *unimodal* (or *single-peak*) if there is only one peak on **g**. (3) *Fewer-peak representation* (*FPR*): Represent **f** by a functional curve with at most a given number k of peaks. It is interesting to note that a nondecreasing curve and a nonincreasing curve of size n each can sum up to form a functional curve **f** with $O(n)$ peaks (e.g., see Fig. 2). The error measure we use in this paper is the *uniform* error metric, also known as the L_∞ metric.

We are concerned with several versions of these problems. For the UDPR problem, its representation complexity (i.e., the number of curves in the representation) is always 2. We consider: (1) the *feasibility* version, which seeks to decide whether an uphill–downhill pair representation is feasible subject to a given bound ϵ on the representation error, and (2) the min-ϵ version, which aims to minimize the representation error ϵ^* among all feasible uphill–downhill pair representations. For the UR problem, the representation complexity is the number of unimodal curves in the representation. For the FPR problem, the representation complexity is the number of peaks on the sought curve. For each of the UR and FPR problems, we consider: (1) the min-k version, which minimizes the representation complexity k^* subject to a given error bound ϵ, and (2) the min-ϵ version, which minimizes the representation error ϵ^* subject to a given bound k on the representation complexity.

For all these problems, we require that **f** and all the simpler functional curves involved be *nonnegative* (i.e., on or above the x-axis). This nonnegativeness requirement is justified by real applications discussed later. Note that this requirement actually makes the problems more theoretically interesting. We should point out that without the above nonnegativeness constraint, some curve representation problems become much easier to solve. For example, for the UDPR problem, if without the nonnegativeness constraint, it is commonly known that one can always find an uphill curve by summing the parts of the curve **f** with positive first derivative and a downhill curve by summing the parts of **f** with negative first derivative such that the sum of these two curves is exactly **f**. Further, in many statistics applications, the data are all nonnegative, e.g., the ages of people in a country, the precipitation of an area, the household income of a state, the employment rate of a time period, etc.

1.2 Motivations and Related Work

Representing a curve by certain classes of structurally simpler curves is a basic problem that is of theoretical value and practical applicability. Motivated by applications in data mining [15, 16], Chun et al. gave a linear time algorithm [11] for approximating a piecewise linear curve by a single unimodal curve, under the L_2 error measure. In [9], Chun et al. studied an extended case in which the approximating function has k peaks, for a given number k, under the L_p error measure. This problem is similar to our FPR min-ϵ problem except that our error measure is different. The algorithm in [9] computes an optimal solution in $O(km^2 + nm \log n)$ time, where m is the number of peaks on the input curve. In addition, an $O(n \log n)$ time algorithm for computing an optimal unimodal function to approximate a piecewise linear curve under the L_p error measure is also given [9]. As shown in [9, 11, 15, 16], the algorithms above are applicable to certain data mining problems. Motivated by applications in statistics [20, 24, 26], Stout [25] considered the unimodal regression problem, aiming to approximate a set of n points by a unimodal step function. He gave three algorithms with time bounds $O(n \log n)$, $O(n)$, and $O(n)$ for the problem under L_1, L_2, and L_∞ error measures, respectively. Another related problem is the isotonic regression problem of approximating a set of points by a nondecreasing step function under the L_p error measure. This problem has been studied considerably in the statistics area (see [20, 24–26] for more details). A one-peak problem in high dimension was studied in [10]. Chen et al. [6] considered the problem of approximating a function on a d-D voxel grid by one unimodal function under the L_2 error metric.

In addition to applications in data mining and statistics, our studies are also motivated by a dose decomposition problem in intensity-modulated radiation therapy (IMRT). IMRT is a modern cancer treatment technique aiming to deliver a prescribed conformal radiation dose to a target tumor while sparing the surrounding normal tissue and critical structures [30, 31]. A prescribed dose function (or curve) $\mathbf{f}$ normally can be made into a piecewise linear form. In the rotational delivery approach [29] (also called dynamic IMRT), a prescribed dose function $\mathbf{f}$ is delivered by repeatedly rotating the radiation source around the patient. In each rotation (called a *path*), a portion of the prescribed dose $\mathbf{f}$ is delivered in a continuous manner. A unimodal dose function can be delivered by a path *smoothly* and *accurately*. Thus, it is desirable to exactly or approximately represent an arbitrary dose curve $\mathbf{f}$ by (the sum of) a minimum set of unimodal curves, for a fast and accurate delivery of the prescribed dose.

In the IMRT settings, Bansal et al. [1] and Chen et al. [7] studied the shape rectangularization problems, which approximate a piecewise linear curve by the sum of a minimum set of constant window functions (or rectangular blocks). (A constant window function $W(\cdot)$ is defined on an interval I such that $W(x)$ is a fixed value $h > 0$ for any $x \in I$ and is 0 otherwise.) Since the shape rectangularization problems are NP-hard [7, 8] (in fact, APX-hard [1]), approximation algorithms were given, and some special cases were solved optimally [1, 7]. Note that constant window functions are a more restricted form of unimodal functions.

Various curve approximation problems have been studied extensively in computational geometry and other applied areas such as cartography, databases, geographic

Table 1 Summary of our results: m is the number of peaks in the input function $\mathbf{f}$ of size n

	UDPR		UR		FPR	
	feasibility	min-ϵ	min-k	min-ϵ	min-k	min-ϵ
Running time	$O(n)$	$O(n)$	$O(n)$	$O(n+m\log m)$	$O(n)$	$O(n)$

information systems, image processing, machine learning, and numerical computing. However, most curve approximation problems seek to simplify a given curve by another "simpler" curve (e.g., with fewer line segments) under certain error criteria (e.g., see [2, 3, 5, 18, 19, 21, 23, 27, 28]). In contrast, although the curve representation problems studied in this paper (which use the *sum of a set of structurally simpler curves* to approximate a given curve) are of mathematical interest and importance, to our best knowledge, not much previous work on these problems has been found in the literature.

1.3 Our Contributions

Based on new geometric observations, we develop efficient algorithmic techniques for various versions of the curve representation problems. For the UDPR problem, we give $O(n)$ time algorithms for both its feasibility version and min-ϵ version. For the UR problem, we present an $O(n)$ time algorithm for its min-k version, and an $O(n+m\log m)$ time algorithm for its min-ϵ version, where $m<n$ is the number of peaks on $\mathbf{f}$. For the FPR problem, our min-k and min-ϵ algorithms both take $O(n)$ time. Our results are summarized in Table 1.

Given an error $\epsilon>0$, our UDPR feasibility algorithm first computes a key structure called *characteristic curve*, which is crucial to identifying the best possible uphill–downhill pair representations. Once the characteristic curve is available, the feasibility can be decided easily. For the UDPR min-ϵ problem, a "natural" way to tackle it is to make use of the UDPR feasibility algorithm. Based on geometric observations, one can determine $O(n^2)$ "critical" errors each of which may cause a *topological change* of the characteristic curve. It is the topological changes of the characteristic curve that affect the feasibility of the UDPR problem. Consequently, the optimal error ϵ^* can be obtained by using the UDPR feasibility algorithm as a search engine. Although there are $O(n^2)$ errors, they can be represented implicitly. An interesting technique, called *binary search on sorted arrays*, can then be applied to find ϵ^* in $O(n\log n)$ time. However, we can do better. By exploiting the geometric structures, we manage to identify a subset of no more than n critical errors. Thus using a prune and search approach, we obtain the optimal error ϵ^* in only $O(n)$ time.

Our UR algorithms are hinged on several key ideas. Interestingly, based on geometric insights, we model the UR problem in a way that a series of UDPR problems needs to be solved. Thus, the UDPR feasibility algorithm is utilized as a subroutine for solving the UR min-k problem. The UR min-k algorithm works in a greedy fashion, in $O(n)$ time. For the UR min-ϵ problem, like the UDPR min-ϵ algorithm, we can find a set S of possible errors which may cause changes to the minimum number of unimodal curves needed to represent $\mathbf{f}$, and $|S|=O(n^4)$. To solve the problem efficiently, our strategy is to first prune the error set S to a subset S' of size $O(n^2)$, by using the UR min-k algorithm as the search engine combined with the technique of binary search on sorted arrays. Note that both S and S' are represented and stored implicitly. Next, we design an efficient data structure that, after a linear time preprocessing, can report each relevant error in S' in $O(1)$ time. Consequently, the optimal error ϵ^* can be obtained efficiently, in actually $O(n + m \log m)$ time, where $m < n$ is the number of peaks on the input curve $\mathbf{f}$.

For the FPR problem, based on its geometry, our $O(n)$ time algorithm solves the min-k version in a greedy fashion. For its min-ϵ version, as our UDPR min-ϵ algorithm, we determine in $O(n)$ time the $O(n)$ critical errors which may cause changes to the minimum number of peaks on the optimal representation curve for $\mathbf{f}$. Consequently, the problem is solvable in $O(n)$ time. Note that for any FPR solution, by truncating it properly, we can always obtain a feasible solution for the corresponding UR problem but it is not necessarily an optimal solution.

It might be tempting to seek an $O(n)$ time UR min-ϵ algorithm by applying similar ideas as those for our UDPR and FPR min-ϵ algorithms. However, so far neither one works for us. The reason is that in either the UDPR or FPR case, each critical error can affect the optimal solution only in a "local" manner, while in the UR case, an error can affect the optimal solution "globally". For example, in the FPR min-ϵ problem, based on our geometric observations, for each peak on the current optimal representation curve, we can determine the error value δ such that when the allowed error $\epsilon \geq \delta$, the peak will disappear from the curve without affecting other peaks on the current representation curve (see Sect. 4 for more details). However, we are not able to do this for the UR case; in other words, we do not know when a unimodal curve in the current representation curve set will not be necessary for representing $\mathbf{f}$ without affecting other unimodal curves in the curve set. A faster UR min-ϵ solution might require more powerful geometric structures (if any).

Although the problems studied in this paper can be viewed as curve approximation problems (e.g., see [2, 3, 5, 18, 19, 21, 23, 27, 28]), we are not aware of any previous curve approximation results that are based on our key geometric structures and observations such as the characteristic curve and critical errors. It is also interesting to note that while the unimodal representation problems are nearly linear time solvable, in contrast, the shape rectangularization problems, which can be viewed as a more restricted case of the unimodal representation problems, are NP-hard [1, 7, 8].

The rest of the paper is organized as follows. In Sect. 2, we consider the uphill–downhill pair representation problems. In Sect. 3, we solve the unimodal representation problems. The fewer-peak representation problems are studied in Sect. 4.

2 The Uphill–Downhill Pair Representation Problem

In this section, we study the following *uphill–downhill pair representation* (*UDPR*) problem: Given a piecewise linear functional curve $\mathbf{f} = (f_1, f_2, \ldots, f_n)$ ($n \geq 2$) and an error bound $\epsilon \geq 0$, find a pair of piecewise linear curves $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ and $\mathbf{z} = (z_1, z_2, \ldots, z_n)$, such that (1) $|y_i + z_i - f_i| \leq \epsilon$ for every $1 \leq i \leq n$, (2) $y_1 \leq y_2 \leq \cdots \leq y_n$, (3) $z_1 \geq z_2 \geq \cdots \geq z_n$, and (4) $\mathbf{y}$ and $\mathbf{z}$ are both nonnegative. If constraint (4) is removed, i.e., the sought curves need not be nonnegative, then we call the

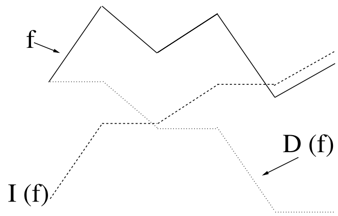

Fig. 3 Illustrating the profile curves $I(\mathbf{f})$ and $D(\mathbf{f})$ of the curve $\mathbf{f}$

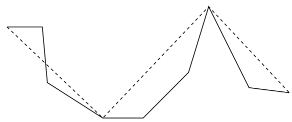

Fig. 4 Illustrating SK($\mathbf{f}$) (the *dashed curve*) of $\mathbf{f}$ (the *solid curve*)

corresponding problem the *relaxed UDPR problem*. Interestingly, our solutions for the UDPR problems are used as a subroutine for solving the UR problems in Sect. 3.

2.1 Preliminaries

We first define some notations used throughout the paper. Given $\mathbf{f} = (f_1, f_2, \ldots, f_n)$, we define a nondecreasing (uphill) piecewise linear functional curve $I(\mathbf{f}) = (I(f_1), I(f_2), \ldots, I(f_n))$ and a nonincreasing (downhill) piecewise linear functional curve $D(\mathbf{f}) = (D(f_1), D(f_2), \ldots, D(f_n))$ as follows: $I(f_1) = 0$, $I(f_i) = I(f_{i-1}) + \max\{f_i - f_{i-1}, 0\}$ for $2 \le i \le n$; $D(f_1) = f_1$, $D(f_i) = D(f_{i-1}) - \max\{f_{i-1} - f_i, 0\}$ for $2 \le i \le n$. Essentially, $I(\mathbf{f})$ is the curve that starts at $I(f_1) = 0$ and increases by the same amount as that from f_{i-1} to f_i if $f_i > f_{i-1}$ and stays the same otherwise; $D(\mathbf{f})$ starts at $D(f_1) = f_1$ and decreases by the same amount as that from f_{i-1} to f_i if $f_{i-1} > f_i$ and stays the same otherwise. Figure 3 shows an example. We call these two curves $I(\mathbf{f})$ and $D(\mathbf{f})$ the *profile curves* of $\mathbf{f}$. Observe that since $f_i = I(f_i) + D(f_i)$ for each $1 \le i \le n$ (this can be easily proved by induction), the profile curves of $\mathbf{f}$ form a solution for the relaxed UDPR problem of $\mathbf{f}$ with any error $\epsilon \ge 0$. In Fig. 2, the two curves $\mathbf{y}$ and $\mathbf{z}$ form a feasible solution for the *relaxed* UDPR problem on $\mathbf{f}$ and $\epsilon = 0$, but not for the UDPR problem if the x-axis passes through the point b instead of the point a.

For a curve $\mathbf{f} = (f_1, f_2, \ldots, f_n)$, similar to the peak definition, we call f_i a *valley* if (1) $i = 1$ and there is a j with $1 < j \le n$ such that $f_1 = \cdots = f_{j-1} < f_j$, or (2) $1 < i < n$, $f_{i-1} > f_i$ and there is a j with $i < j \le n$ such that $f_i = \cdots = f_{j-1} < f_j$, or (3) $i = n$ and there is a j with $1 \le j \le n-1$ such that $f_j > f_{j+1} = \cdots = f_n$. Clearly, there is exactly one valley (resp., peak) between any two consecutive peaks (resp., valleys) on a curve. For a curve $\mathbf{f}$, we define its *skeleton* SK($\mathbf{f}$) by connecting each peak (resp., valley) to its right side consecutive valley (resp., peak) with a line segment (see Fig. 4). A curve $\mathbf{f}'$ is called a *skeleton curve* if each f'_i is either a peak or a valley and a *general curve* otherwise. Below we sometimes "abuse" the notation by denoting SK($\mathbf{f}$) by $\mathbf{f}$. Note that the following analysis on skeleton curve can also be done on a general curve and the reason we define the skeleton curve is for simplicity of exposition and to derive a tight algorithm time bound.

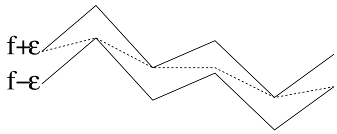

Fig. 5 Illustrating the characteristic curve $R(\mathbf{f},\epsilon)$ (the *dashed one*). The two *solid curves* are $\mathbf{f}+\epsilon$ and $\mathbf{f}-\epsilon$

Given a skeleton curve $\mathbf{f}=(f_1,f_2,\dots,f_n)$ and $\epsilon\geq 0$, the *characteristic curve* of $\mathbf{f}$ and ϵ, denoted by $R(\mathbf{f},\epsilon)$, is defined as $R(\mathbf{f},\epsilon)=(R_1,R_2,\dots,R_n)$, where $R_1=f_1+\epsilon$, R_i is equal to $f_i-\epsilon$ if $R_{i-1}<f_i-\epsilon$, R_i is $f_i+\epsilon$ if $R_{i-1}>f_i+\epsilon$, and $R_i=R_{i-1}$ otherwise (see Fig. 5). The skeleton curve plays an important role in our following algorithms.

2.2 The Feasibility of the UDPR Problem

In this section, we study the feasibility of the UDPR problem. For $\epsilon\geq 0$, we say that a curve $\mathbf{f}=(f_1,f_2,\dots,f_n)$ (with $n\geq 2$) is ϵ*-UDP-representable* if the UDPR problem on $\mathbf{f}$ and ϵ is feasible. We first focus on the UDPR feasibility algorithm for a skeleton curve. We will show later that $\mathbf{f}$ is ϵ-UDP-representable if and only if its skeleton SK($\mathbf{f}$) is ϵ-UDP-representable, and the solution for $\mathbf{f}$ (resp., SK($\mathbf{f}$)) can be obtained in linear time once we have the solution for SK($\mathbf{f}$) (resp., $\mathbf{f}$). The UDPR feasibility of a skeleton curve can be determined by the following lemma.

Lemma 1 *Given a skeleton curve $\mathbf{f}=(f_1,f_2,\dots,f_n)$ and $\epsilon>0$, suppose $R(\mathbf{f},\epsilon)$ is its characteristic curve. Then $\mathbf{f}$ is ϵ-UDP-representable if and only if $D(R_n)\geq 0$. Moreover, if $\mathbf{f}$ is ϵ-UDP-representable, then the profile curves of $R(\mathbf{f},\epsilon)$ form a UDPR solution.*

Proof If $D(R_n)\geq 0$, then since $I(R_1)=0$, the profile curves $I(R(\mathbf{f},\epsilon))$ and $D(R(\mathbf{f},\epsilon))$ of $R(\mathbf{f},\epsilon)$ are both nonnegative. Due to the facts that $I(R(\mathbf{f},\epsilon))+D(R(\mathbf{f},\epsilon))=R(\mathbf{f},\epsilon)$ and $R(\mathbf{f},\epsilon)$ is bounded between the two curves $\mathbf{f}+\epsilon$ and $\mathbf{f}-\epsilon$ (e.g., see Fig. 5), the two profile curves of $R(\mathbf{f},\epsilon)$ form a feasible UDPR solution for $\mathbf{f}$ and ϵ.

Suppose $\mathbf{f}$ is ϵ-UDP-representable. Let an uphill curve $\mathbf{y}=(y_1,y_2,\dots,y_n)$ and a downhill curve $\mathbf{z}=(z_1,z_2,\dots,z_n)$ form a feasible UDPR solution for $\mathbf{f}$ and ϵ. To show $D(R_n)\geq 0$, since $z_n\geq 0$, it is sufficient to prove $D(R_n)\geq z_n$. We claim that, for any $1\leq i\leq n$, $I(R_i)\leq y_i$ and $D(R_i)\geq z_i$.

We prove the claim by induction. Note that $|y_i+z_i-f_i|\leq\epsilon$ for any $1\leq i\leq n$. For $i=1$, since $y_1\geq 0$, we have $I(R_1)=0\leq y_1$ and $D(R_1)=f_1+\epsilon\geq f_1+\epsilon-y_1\geq z_1$. For $i\geq 2$, by induction, we have $I(R_{i-1})\leq y_{i-1}$ and $D(R_{i-1})\geq z_{i-1}$. Since $y_i\geq y_{i-1}$, $z_i\leq z_{i-1}$, and $y_i+z_i\geq f_i-\epsilon$, we have $y_i\geq\max\{y_{i-1},f_i-\epsilon-z_{i-1}\}$. On the other hand, by the definition of the $I(\cdot)$ curve, $I(R_i)=\max\{I(R_{i-1}),I(R_{i-1})+R_i-R_{i-1}\}=\max\{I(R_{i-1}),R_i-D(R_{i-1})\}$. We claim $I(R_i)=\max\{I(R_{i-1}),f_i-\epsilon-D(R_{i-1})\}$. To see why, if $I(R_{i-1})\geq f_i-\epsilon-D(R_{i-1})$, then clearly $I(R_i)=I(R_{i-1})$. Otherwise ($I(R_{i-1})<f_i-\epsilon-D(R_{i-1})$, which means $R_{i-1}<f_i-\epsilon$), we have $R_i=f_i-\epsilon$, and hence the claim still holds. Due to the above claim and by the induction hypothesis (i.e., $I(R_{i-1})\leq y_{i-1}$ and $D(R_{i-1})\geq z_{i-1}$), we have

$I(R_i) = \max\{I(R_{i-1}), f_i - \epsilon - D(R_{i-1})\} \le \max\{y_{i-1}, f_i - \epsilon - z_{i-1}\} \le y_i$. Thus we prove $I(R_i) \le y_i$. Similarly, we can prove $D(R_i) \ge z_i$. □

Given a skeleton curve $\mathbf{f}$ and $\epsilon \ge 0$, since the characteristic curve $R(\mathbf{f}, \epsilon)$ and its profile curves can all be computed in linear time, by Lemma 1, we immediately have the following result.

Lemma 2 *The UDPR feasibility problem on a skeleton curve* $\mathbf{f}$ *and* $\epsilon \ge 0$ *is solvable in* $O(n)$ *time.*

The problem on an arbitrary piecewise linear functional curve can be handled by the next lemma.

Lemma 3 *For a general piecewise linear functional curve* $\mathbf{f} = (f_1, f_2, \dots, f_n)$ *and* $\epsilon \ge 0$, $\mathbf{f}$ *is* ϵ*-UDP-representable if and only if* $\mathrm{SK}(\mathbf{f})$ *is* ϵ*-UDP-representable. Furthermore, given a feasible solution for* $\mathrm{SK}(\mathbf{f})$ (*resp.*, $\mathbf{f}$), *a feasible solution for* $\mathbf{f}$ (*resp.*, $\mathrm{SK}(\mathbf{f})$) *can be obtained in* $O(n)$ *time.*

Proof In the xy-plane, let x_i be the x-coordinate for f_i, $1 \le i \le n$, i.e., the point (x_i, f_i) is the ith vertex of $\mathbf{f}$. The vertices of $\mathrm{SK}(\mathbf{f})$ are a subset of the vertices of $\mathbf{f}$. Let $|\mathrm{SK}(\mathbf{f})| = m$. For the ith vertex of $\mathrm{SK}(\mathbf{f})$, $1 \le i \le m$, let $l(i)$ denote the index of the same vertex on $\mathbf{f}$. If $\mathbf{f}$ is ϵ-UDP-representable, then its UDPR feasible solution can clearly yield a feasible solution for $\mathrm{SK}(\mathbf{f})$.

If $\mathrm{SK}(\mathbf{f})$ is ϵ-UDP-representable, then by Lemma 1, the profile curves of $R(\mathrm{SK}(\mathbf{f}), \epsilon)$ form a feasible solution for $\mathrm{SK}(\mathbf{f})$. Let these profile curves be an uphill curve $\mathbf{y} = (y_1, y_2, \dots, y_m)$ and a downhill curve $\mathbf{z} = (z_1, z_2, \dots, z_m)$. A feasible solution for $\mathbf{f}$ consisting of an uphill curve $\mathbf{y}' = (y'_1, y'_2, \dots, y'_n)$ and a downhill curve $\mathbf{z}' = (z'_1, z'_2, \dots, z'_n)$ can be obtained in $O(n)$ time, as follows.

For each $1 \le i \le m-1$, by the definition of $\mathrm{SK}(\mathbf{f}) = (f_{l(1)}, f_{l(2)}, \dots, f_{l(m)})$, either $f_{l(i)} > f_{l(i+1)}$ or $f_{l(i)} < f_{l(i+1)}$. WLOG, we assume it is the former case (the latter case can be handled similarly). The case with $l(i+1) = l(i) + 1$ is trivial. In the following, we assume $l(i+1) > l(i) + 1$. Let the characteristic curve $R(\mathrm{SK}(\mathbf{f}), \epsilon)$ be $(R_1, R_2, \dots, R_m)$. Since $f_{l(i)} > f_{l(i+1)}$, the portion of $\mathbf{f}$ from $f_{l(i)}$ to $f_{l(i+1)}$ is nonincreasing, and thus we have $R_i \ge R_{i+1}$. By the definition of profile curves, $y_{i+1} = y_i$ and $z_{i+1} = z_i - (R_i - R_{i+1})$. The values y'_j and z'_j for all $l(i) \le j \le l(i+1)$ can be obtained as follows. First, let $y'_{l(i)} = y_i, z'_{l(i)} = z_i$ and $y'_{l(i+1)} = y_{i+1}, z'_{l(i+1)} = z_{i+1}$. Since $y_i + z_i = R_i$ and $y_{i+1} + z_{i+1} = R_{i+1}$, we have $y'_{l(i)} + z'_{l(i)} = R_i \in [f_{l(i)} - \epsilon, f_{l(i)} + \epsilon]$ and $y'_{l(i+1)} + z'_{l(i+1)} = R_{i+1} \in [f_{l(i+1)} - \epsilon, f_{l(i+1)} + \epsilon]$. For any j with $l(i) < j < l(i+1)$, set $y'_j = y'_{l(i)}$, which means the $\mathbf{y}'$ values do not change from $y'_{l(i)}$ to $y'_{l(i+1)}$. If $R_i = R_{i+1}$, then $z'_{l(i)} = z'_{l(i+1)}$. We set $z'_j = z'_{l(i)}$ for any $l(i) < j < l(i+1)$. Due to $f_{l(i)} \ge f_j \ge f_{l(i+1)}$ for any $l(i) < j < l(i+1)$ and $y'_j + z'_j = R_i = R_{i+1}$, it must be $f_j - \epsilon \le y'_j + z'_j \le f_j + \epsilon$ for any $l(i) < j < l(i+1)$. If $R_i > R_{i+1}$, let $L = R_i - f_{l(i)}$, and thus $|L| \le \epsilon$. (Note that when $R_i > R_{i+1}$, it must be $R_{i+1} = f_{l(i+1)} + \epsilon$.) Going from $f_{l(i)}$ to $f_{l(i+1)}$ on $\mathbf{f}$, suppose t is the smallest index such that $f_t + L < R_{i+1}$; then we set $z'_j = f_j + L$ for $l(i) < j < t$ and $z'_j = z'_{l(i+1)}$ for $t \le j < l(i+1)$. Figure 6 shows an example in which the fourth vertical line from

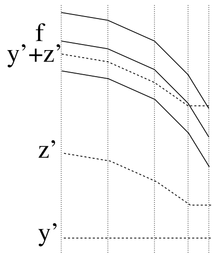

Fig. 6 Illustrating the *three* (*dashed*) *curves* $y' + z'$, z', and y'. The three *solid curves* are $\mathbf{f} + \epsilon$, $\mathbf{f}$, and $\mathbf{f} - \epsilon$

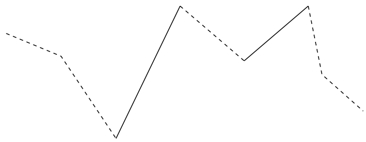

Fig. 7 The "*downhill*" portions of a *curve*

the left stands for t. It is easy to see that the z' values from $l(i)$ to $l(i+1)$ are non-increasing and for any $l(i) < j < l(i+1)$, it must be $f_j - \epsilon \le y'_j + z'_j \le f_j + \epsilon$. In this way, a feasible UDPR solution for $\mathbf{f}$ can be obtained in $O(n)$ time, which proves the lemma. □

From Lemmas 2 and 3, we have the following theorem.

Theorem 1 *The UDPR feasibility problem on an arbitrary piecewise linear functional curve* $\mathbf{f}$ *of size* n *and* $\epsilon \ge 0$ *is solvable in* $O(n)$ *time.*

2.3 The min-ϵ Version of the UDPR Problem

In this section, we consider the UDPR min-ϵ problem, seeking the minimum possible error ϵ for $\mathbf{f}$ to be ϵ-UDP-representable. In light of Lemma 3, we only need to develop an algorithm for the skeleton of $\mathbf{f}$. Let ϵ^* denote the sought minimum error. At first sight, one might attempt to solve the UDPR min-ϵ problem by utilizing the result in Theorem 1 and performing binary search for ϵ^*. But that would lead to a superlinear time solution. Our UDPR min-ϵ algorithm takes $O(n)$ time.

2.3.1 Useful Geometric Observations

Given a skeleton curve $\mathbf{f} = (f_1, f_2, \dots, f_n)$ and $\epsilon \ge 0$, by Lemma 1, $\mathbf{f}$ is ϵ-UDP-representable if and only if $D(R_n) \ge 0$. By the definition of the profile curves, we have $D(R_n) = R_1 - \sum_{i=2}^{n} \max\{0, R_{i-1} - R_i\}$ and $R_1 = f_1 + \epsilon$. For a general functional curve $\mathbf{h} = (h_1, h_2, \dots, h_n)$, we define $\mathcal{H}(\mathbf{h})$ to be $\sum_{i=2}^{n} \max\{0, h_{i-1} - h_i\}$. Geometrically, the value of $\mathcal{H}(\mathbf{h})$ is the sum of the "height drops" of all the "downhill" portions of the curve $\mathbf{h}$ (see Fig. 7). Then we have $D(R_n) = f_1 + \epsilon - \mathcal{H}(R(\mathbf{f}, \epsilon))$.

On the characteristic curve $R(\mathbf{f}, \epsilon) = (R_1, R_2, \dots, R_n)$, we call R_i an *R-peak* if R_i is a peak on $R(\mathbf{f}, \epsilon)$ with $1 < i < n$. Thus R_1 and R_n cannot be R-peaks. For each R-peak R_i, we define its *allied R-valley* to be R_j, where R_j is the first valley on $R(\mathbf{f}, \epsilon)$ to the right of R_i, i.e., $j = \min\{t \mid t > i \text{ and } R_t \text{ is a valley}\}$. An R-peak R_i

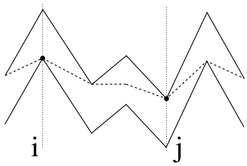

Fig. 8 Illustrating an allied pair (R_i, R_j) (*black points*) on $R(\mathbf{f}, \epsilon)$ (the *dashed curve*)

and its allied R-valley R_j form an *allied pair* (R_i, R_j) (see Fig. 8). The following observation is based on the geometric properties of the characteristic curve.

Observation 1 *For any $\epsilon \geq 0$, if R_i is an R-peak on $R(\mathbf{f}, \epsilon)$, then $R_i = f_i - \epsilon$ and f_i is a peak on $\mathbf{f}$; if R_j is the allied R-valley of the R-peak R_i on $R(\mathbf{f}, \epsilon)$, then $R_j = f_j + \epsilon$ and f_j is a valley on $\mathbf{f}$.*

We name the sequence of the allied pairs of $R(\mathbf{f}, \epsilon)$ (from left to right) the *topology* of $R(\mathbf{f}, \epsilon)$.

Lemma 4 *Given a skeleton curve $\mathbf{f}$ and an error $\epsilon \geq 0$, if both the curves $R(\mathbf{f}, \epsilon)$ and $R(\mathbf{f}, \epsilon + \Delta\epsilon)$ has the same topology for a value $\Delta\epsilon$, then $\mathcal{H}(R(\mathbf{f}, \epsilon + \Delta\epsilon)) = \mathcal{H}(R(\mathbf{f}, \epsilon)) - 2\Delta\epsilon \cdot \alpha$, where α is the number of allied pairs on $R(\mathbf{f}, \epsilon)$ (as well as on $R(\mathbf{f}, \epsilon + \Delta\epsilon)$).*

Proof WLOG, assume $\Delta\epsilon \geq 0$. Let $R(\mathbf{f}, \epsilon) = (R_1, R_2, \ldots, R_n)$ and $R(\mathbf{f}, \epsilon + \Delta\epsilon) = (R'_1, R'_2, \ldots, R'_n)$. Denote the set of the allied pairs on $R(\mathbf{f}, \epsilon)$ by S. For each pair $(R_i, R_j) \in S$, since both the curves have the same topology, (R'_i, R'_j) is an allied pair on $R(\mathbf{f}, \epsilon + \Delta\epsilon)$. Suppose R_i is the first R-peak on $R(\mathbf{f}, \epsilon)$. Let C denote the sum of the "height drops" of the "downhill" portion from R_1 to R_{i-1} (note that $C > 0$ if and only if f_1 is a peak on $\mathbf{f}$). Since R_1 is at $f_1 + \epsilon$ and the downhill portion for C ends at $f_t + \epsilon$ for some $1 < t < i$, C is a constant as long as the topology of $R(\mathbf{f}, \epsilon)$ does not change. By the allied pair definition, we have $\mathcal{H}(R(\mathbf{f}, \epsilon)) = C + \sum_{(R_i, R_j) \in S}(R_i - R_j)$ and $\mathcal{H}(R(\mathbf{f}, \epsilon + \Delta\epsilon)) = C + \sum_{(R'_i, R'_j) \in S}(R'_i - R'_j)$. Therefore, when the error changes from ϵ to $\epsilon + \Delta\epsilon$, since the topology does not change, for each allied pair (R_i, R_j), by Observation 1, we have (1) $R_i = f_i - \epsilon$ and $R'_i = f_i - (\epsilon + \Delta\epsilon)$, and (2) $R_j = f_j + \epsilon$ and $R'_j = f_j + (\epsilon + \Delta\epsilon)$. Thus $R_i - R'_i = R'_j - R_j = \Delta\epsilon$, which yields the lemma due to $|S| = \alpha$. □

The above lemma implies that if the topology of $R(\mathbf{f}, \epsilon)$ does not change for $\epsilon \in [\epsilon_1, \epsilon_2]$, then $\mathcal{H}(R(\mathbf{f}, \epsilon))$ is a continuous decreasing linear function in that interval. Denote by $\mathcal{M}(\epsilon)$ the number of allied pairs on $R(\mathbf{f}, \epsilon)$. Thus $\mathcal{M}(0)$ is the number of allied pairs on $\mathbf{f}$ (when $\epsilon = 0$, $R(\mathbf{f}, \epsilon) = \mathbf{f}$). Note that as ϵ increases from 0 to ∞, at some values of ϵ, the topology of $R(\mathbf{f}, \epsilon)$ will change and the value of $\mathcal{M}(\epsilon)$ will decrease by some integer $t \geq 1$. When ϵ is large enough, $\mathcal{M}(\epsilon)$ becomes zero and never decreases any more. Thus, $\mathcal{M}(\epsilon)$ is a nonincreasing step function (see Fig. 9), and the number of steps is at most $\mathcal{M}(0)$. Suppose the ith "step" of $\mathcal{M}(\epsilon)$ is defined on the interval $[\epsilon_i, \epsilon_{i+1})$; then we call ϵ_i a *critical* error if $i \geq 1$ ($\epsilon_1 = 0$ is not considered to be a critical error). Formally, ϵ' is a critical error if and only if

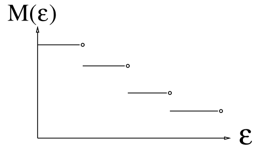

Fig. 9 $\mathcal{M}(\epsilon)$ is a step function

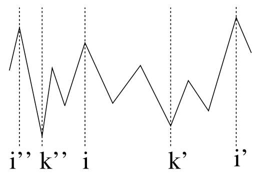

Fig. 10 Illustrating $\epsilon_i' = (f_i - f_{k'})/2$ for the peak f_i

$\mathcal{M}(L(\epsilon')) - \mathcal{M}(\epsilon') > 0$, where $L(\epsilon')$ is a value less than ϵ' but infinitely close to it. We use a multi-set E to denote the set of all critical errors: For each critical error ϵ', if $\mathcal{M}(L(\epsilon')) - \mathcal{M}(\epsilon') = t \geq 1$, then E contains t copies of ϵ'. Thus $|E|$ is exactly equal to $\mathcal{M}(0)$.

From a geometric point of view, $R(\mathbf{f}, \epsilon)$ changes its topology only when a peak of the curve $\mathbf{f} - \epsilon$ "touches" some point of a horizontal segment of $R(\mathbf{f}, \epsilon)$ starting at a valley of $\mathbf{f} + \epsilon$. (Since each horizontal segment of $R(\mathbf{f}, \epsilon)$ starts at a valley of $\mathbf{f} + \epsilon$, it cannot be touched by another valley of $\mathbf{f} + \epsilon$.) When a peak $f_i - \epsilon$ of $\mathbf{f} - \epsilon$ touches a horizontal segment of $R(\mathbf{f}, \epsilon)$ starting at a valley $f_j + \epsilon$ of $\mathbf{f} + \epsilon$, we have $f_i - \epsilon = f_j + \epsilon$, implying $\epsilon = \frac{|f_i - f_j|}{2}$. Let $E' = \{|f_i - f_j|/2 \mid$ for any peak f_i and valley f_j on $\mathbf{f}\}$. Then clearly, the critical error set E is a subset of E'. Thus we have the following lemma.

Lemma 5 *Given a skeleton curve* $\mathbf{f}$, *the function* $\mathcal{G}(\epsilon) = f_1 + \epsilon - \mathcal{H}(R(\mathbf{f}, \epsilon))$ (*i.e.*, $\mathcal{G}(\epsilon) = D(R_n)$) *is a continuous increasing piecewise linear function for* $\epsilon \geq 0$. *More specifically, the interval* $[0, +\infty)$ *for* ϵ *can be partitioned into* $|E'| + 1$ *sub-intervals by the elements in* E', *such that in each such sub-interval*, $\mathcal{G}(\epsilon)$ *is an increasing linear function of* ϵ.

2.3.2 The Algorithm

Our algorithm first determines the multi-set E explicitly and then computes ϵ^*.

Let $P(\mathbf{f})$ denote the set of indices of all peaks on $\mathbf{f}$ except f_1 and f_n. When $\epsilon = 0$, since $R(\mathbf{f}, \epsilon)$ is the same as $\mathbf{f}$, R_i is an R-peak on $R(\mathbf{f}, \epsilon)$ if and only if $i \in P(\mathbf{f})$. Thus $|P(\mathbf{f})| = \mathcal{M}(0)$. For each $i \in P(\mathbf{f})$, let $i' = \min\{t \mid i < t \leq n + 1, f_t > f_i\}$ (with $f_{n+1} = +\infty$); in other words, $f_{i'}$ is the leftmost peak to the right of f_i that is *larger than* f_i, or $i' = n + 1$ if there is no such peak on $\mathbf{f}$. Let $i'' = \max\{t \mid 0 \leq t < i, f_t \geq f_i\}$ (with $f_0 = +\infty$), i.e., $f_{i''}$ is the rightmost peak to the left of f_i that is *larger than or equal to* f_i, or $i'' = 0$ if there is no such peak (see Fig. 10). For each $i \in P(\mathbf{f})$, let $f_{k'} = \min\{f_t \mid i < t < i'\}$, $f_{k''} = \min\{f_t \mid i'' < t < i\}$, and $\epsilon_i' = (f_i - \max\{f_{k'}, f_{k''}\})/2$. Figure 10 shows an example. Note that for each ϵ_i', the paring of f_i and $\max\{f_{k'}, f_{k''}\}$ is similar to the paring of extrema in persistent homology [13, 32] (which is an algebraic study of measuring topological features of shapes and of functions), and the value $2\epsilon'_i$ is called *persistence* there. The next lemma is crucial for computing E.

Lemma 6 *For any $\epsilon \geq 0$, $1 < i < n$, R_i is an R-peak on $R(\mathbf{f}, \epsilon)$ if and only if $i \in P(\mathbf{f})$ and $\epsilon < \epsilon'_i$.*

Proof For each $i \in P(\mathbf{f})$, we define i', i'', k', and k'' as above.

If R_i is an R-peak, then by Observation 1, $i \in P(\mathbf{f})$ and $R_i = f_i - \epsilon$. We prove $\epsilon < \epsilon'_i$ by contradiction. Assume $\epsilon \geq \epsilon'_i$. There are two cases to consider: $\epsilon'_i = (f_i - f_{k'})/2$ or $\epsilon'_i = (f_i - f_{k''})/2$. If $\epsilon'_i = (f_i - f_{k'})/2$, suppose R_j is its allied R-valley, then (1) if $i' < n+1$, it must be $j \leq i' - 1$ due to $R_i = f_i - \epsilon$ and $f_{i'} > f_i$; (2) if $i' = n+1$, then obviously $j \leq i' - 1$. Thus in either case, $j \leq i' - 1$. For each t with $i < t \leq j$, we have $f_t - \epsilon < f_i - \epsilon \leq f_t + \epsilon$ due to $f_{k'} \leq f_t < f_i$ and $\epsilon \geq (f_i - f_{k'})/2$. By the definition of the characteristic curve, for every $i < t \leq j$, R_t should be equal to R_i, implying that R_j is not the allied R-valley of R_i, a contradiction. If $\epsilon'_i = (f_i - f_{k''})/2$, then either $i'' > 0$ or $i'' = 0$. In either case, we claim $R_{k''} \geq R_i$. If $i'' > 0$, then due to $f_{i''} \geq f_i$ and $R_i = f_i - \epsilon$, we have $R_{i''} \geq f_{i''} - \epsilon \geq R_i$. If $R_{i''} < f_{k''} + \epsilon$, then $R_{k''} = R_{i''} \geq R_i$ since $f_{k''} = \min\{f_t \mid i'' \leq t \leq k''\}$ and $f_{i''} = \max\{f_t \mid i'' \leq t \leq k''\}$. Similarly, if $R_{i''} \geq f_{k''} + \epsilon$, then by the definition of the curve $R(\mathbf{f}, \epsilon)$ and $f_{k''} = \min\{f_t \mid i'' \leq t \leq k''\}$, it must be $R_{k''} = f_{k''} + \epsilon$. Due to $\epsilon \geq \epsilon'_i = (f_i - f_{k''})/2$ and $R_i = f_i - \epsilon$, we have $R_{k''} \geq R_i$. If $i'' = 0$, then since $R_1 = f_1 + \epsilon$ and $f_{k''} = \min\{f_t \mid 1 \leq t \leq k''\}$, by the definition of $R(\mathbf{f}, \epsilon)$, it must be $R_{k''} = f_{k''} + \epsilon$. Thus $R_{k''} \geq R_i$ still holds due to $\epsilon \geq (f_i - f_{k''})/2$. Since $R_{k''} \geq R_i$, $f_{k''} = \min\{f_t \mid k'' \leq t \leq i\}$, and $f_i = \max\{f_t \mid k'' \leq t \leq i\}$, we have $R_t = R_i$ for any $k'' \leq t < i$. Thus $R_{i-1} = R_i$, a contradiction to the fact that R_i is an R-peak. This proves $\epsilon < \epsilon'_i$.

If $i \in P(\mathbf{f})$ and $\epsilon < \epsilon'_i$, then both $\epsilon < (f_i - f_{k'})/2$ and $\epsilon < (f_i - f_{k''})/2$ hold. Due to $\epsilon < (f_i - f_{k''})/2$, we have $f_{k''} + \epsilon < f_i - \epsilon$. Since $R_{k''} \leq f_{k''} + \epsilon$, we have $R_{k''} < f_i - \epsilon$. Since $f_{k''} = \min\{f_t \mid i'' < t < i\}$ and $f_i > \max\{f_t \mid k'' < t < i\}$, by the definition of characteristic curve, we have $R_{i-1} = \max\{R_{k''}, \max\{f_t - \epsilon \mid k'' < t < i\}\} < f_i - \epsilon = R_i$. Then to finish proving that R_i is an R-peak, it suffices to show that there is a j with $i < j \leq n$ such that $R_i = R_{i+1} = \cdots = R_{j-1} > R_j$. Since $f_i \geq \max\{f_t \mid i \leq t \leq k'\}$, we have $R_i \geq \max\{R_t \mid i \leq t \leq k'\}$. If there is a t with $i < t < k'$ such that $R_i = R_{i+1} = \cdots = R_{t-1} > R_t$, then we are done. Otherwise, it must be $R_i = R_{i+1} = \cdots = R_{k'-1}$. Due to $\epsilon < (f_i - f_{k'})/2$, we have $f_i - \epsilon > f_{k'} + \epsilon$. Since $R_i = f_i - \epsilon$ and $R_{k'} \leq f_{k'} + \epsilon$, we have $R_i = R_{i+1} = \cdots = R_{k'-1} > R_{k'}$. Thus R_i is an R-peak on $R(\mathbf{f}, \epsilon)$. $\square$

In light of the above lemma, the multi-set E can be determined based on the following lemma.

Lemma 7 $E = \{\epsilon'_i \mid$ *for each* $i \in P(\mathbf{f})\}$.

Proof Recall that there are t copies of ϵ' in E if and only if $\mathcal{M}(L(\epsilon')) - \mathcal{M}(\epsilon') = t \geq 1$. Let $S = \{\epsilon'_i \mid \text{for any } i \in P(\mathbf{f})\}$ be a multi-set. Since $|E| = \mathcal{M}(0) = |P(\mathbf{f})| = |S|$, to prove $E = S$, it suffices to show that for each $\epsilon'_i \in S$, if there are t copies of ϵ'_i in S, then there are also t copies of ϵ'_i in E.

For each ϵ'_i, suppose r elements in S are larger than or equal to ϵ'_i. Then the number of elements in S that are larger than $L(\epsilon'_i)$ is r. If we let $\epsilon = L(\epsilon'_i)$, then by Lemma 6, there are r R-peaks on $R(\mathbf{f}, \epsilon)$ (or $\mathcal{M}(L(\epsilon'_i)) = r$). Suppose there are t copies of ϵ'_i in S; then there are $r - t$ elements in S that are larger than ϵ'_i. When $\epsilon = \epsilon'_i$, by Lemma 6, there are $r - t$ R-peaks on $R(\mathbf{f}, \epsilon)$ (or $\mathcal{M}(\epsilon'_i) = r - t$). Since $\mathcal{M}(L(\epsilon'_i)) - \mathcal{M}(\epsilon'_i) = t$, E contains t copies of ϵ'_i. This proves the lemma. □

To compute E explicitly, although the framework and techniques in persistent homology [13, 32] might be applied, we give a simple optimal $O(n)$ time algorithm, as follows. For each $i \in P(\mathbf{f})$, if we know i' and i'', then ϵ'_i can be obtained in $O(1)$ time by a range minimum data structure [17] (with an $O(n)$ time preprocessing). For all $i \in P(\mathbf{f})$, computing i' is essentially the following problem: Given an array $A[1, \ldots, n]$, for each $1 \le i \le n$, find i' that is the index of the first element after $A[i]$ such that $A[i] < A[i']$. This problem can be easily solved in $O(n)$ time. For each $i \in P(\mathbf{f})$, i'' can be computed similarly. Thus E can be obtained in $O(n)$ time.

Consequently, the value ϵ^* can be computed by the following lemma.

Lemma 8 *After E is obtained, ϵ^* can be computed in $O(|E|)$ time.*

Proof Assume that the elements in E are $\epsilon_1 \le \epsilon_2 \le \cdots \le \epsilon_M$, where $M = |E| = \mathcal{M}(0)$ (this assumption is only for analysis since we do not sort them in the algorithm). By Lemma 5, the function $\mathcal{G}(\epsilon) = f_1 + \epsilon - \mathcal{H}(R(\mathbf{f}, \epsilon))$ is increasing, and thus ϵ^* is the unique value with $\mathcal{G}(\epsilon^*) = 0$. By Lemma 4, $\mathcal{G}(0) = f_1 - \mathcal{H}(R(\mathbf{f}, 0))$, $\mathcal{G}(\epsilon_1) = \mathcal{G}(0) + \epsilon_1 + 2M \cdot \epsilon_1$, and $\mathcal{G}(\epsilon_2) = \mathcal{G}(0) + \epsilon_2 + 2M \cdot \epsilon_1 + 2(M-1) \cdot (\epsilon_2 - \epsilon_1)$. Generally, if we let $\epsilon_0 = 0$, then for $1 \le i \le M$, $\mathcal{G}(\epsilon_i) = \mathcal{G}(0) + \epsilon_i + 2\sum_{t=0}^{i-1}(M - t)(\epsilon_{t+1} - \epsilon_t)$. Thus, geometrically, $\mathcal{G}(\epsilon)$ is a piecewise linear concave increasing function whose slope, when $\epsilon \in [\epsilon_i, \epsilon_{i+1})$, is $1 + 2(M - i)$ for any $0 \le i \le M$ (let ϵ_{M+1} be ∞). Note that if the elements in E are already sorted, then it is easy to compute ϵ^* in linear time since each $\mathcal{G}(\epsilon_i)$ can be obtained from $\mathcal{G}(\epsilon_{i-1})$ in $O(1)$ time and $\mathcal{G}(\epsilon)$ is an increasing function. However, as we show below, we can still compute ϵ^* in linear time without sorting the elements in E. Define $h(i, j) = \sum_{t=i}^{j-1}(M - t)(\epsilon_{t+1} - \epsilon_t)$. Then $\mathcal{G}(\epsilon_i) = \mathcal{G}(0) + \epsilon_i + 2h(0, i)$. By a simple deduction, we can get $h(i, j) = \sum_{t=i+1}^{j-1} \epsilon_t + (M - j + 1)\epsilon_j - (M - i)\epsilon_i$. Thus, we can compute the value of $h(i, j)$ in $O(j - i)$ time if we know all the values $\epsilon_i, \epsilon_{i+1}, \ldots, \epsilon_j$ (actually, in the algorithm below, when computing $h(i, j)$, all the values $\epsilon_i, \epsilon_{i+1}, \ldots, \epsilon_j$ are determined by using the selection algorithm [12]). Further, $\mathcal{G}(0)$ can be easily computed in linear time.

To obtain ϵ^*, we do the following: (1) Search in E for the two elements ϵ' and ϵ'' such that ϵ' is the largest element in E with $\mathcal{G}(\epsilon') \le 0$ and ϵ'' is the smallest one with $\mathcal{G}(\epsilon'') > 0$; (2) compute the smallest value $\epsilon^* \in [\epsilon', \epsilon'']$ such that $\mathcal{G}(\epsilon^*) = 0$. In step (1), to find ϵ', a straightforward way is to first sort all elements in E, and then from the smallest element to the largest one, check the value of $\mathcal{G}(\epsilon_i)$ for each ϵ_i. But that takes $O(M \log M)$ time. An $O(M)$ time algorithm, based on prune and search, works as follows. We first use the selection algorithm [12] to find the median $\epsilon_{M/2}$ in E and compute $\mathcal{G}(\epsilon_{M/2})$, for which we need to spend $O(\frac{M}{2})$ time to compute $h(0, \frac{M}{2})$. If $\mathcal{G}(\epsilon_{M/2}) = 0$, then the algorithm stops with $\epsilon^* = \epsilon_{M/2}$. Otherwise, let $E_1 = \{\epsilon_i \mid i < \frac{M}{2}\}$ and $E_2 = \{\epsilon_i \mid i > \frac{M}{2}\}$. If $\mathcal{G}(\epsilon_{M/2}) < 0$, then we continue the same procedure on E_2. Since we already have the value of $h(0, \frac{M}{2})$, when computing $h(0, j)$ for $j > \frac{M}{2}$, we only need to compute $h(\frac{M}{2} + 1, j)$ because $h(0, j) = h(0, \frac{M}{2}) + h(\frac{M}{2} + 1, j)$, which takes $O(j - \frac{M}{2})$ time. If $\mathcal{G}(\epsilon_{M/2}) > 0$, then we continue the same procedure on E_1. Thus the total time for computing ϵ' is $O(M)$. To obtain ϵ'', note that ϵ'' is the smallest element in E that is larger than ϵ', and thus ϵ'' can be found in linear time. Step (2) takes $O(1)$ time since when $\epsilon \in [\epsilon', \epsilon'']$, $\mathcal{G}(\epsilon)$ is a linear function. □

Therefore, we have the following result.

Theorem 2 *The UDPR* min-ϵ *problem on* $\mathbf{f} = (f_1, f_2, \ldots, f_n)$ *can be solved in* $O(n)$ *time*.

3 The Unimodal Representation Problem

In this section, we study both the min-k and min-ϵ versions of the *unimodal representation* (*UR*) problem. Note that all unimodal curves in the representation are required to be nonnegative. In the following, when we say unimodal curve, we mean nonnegative unimodal curve.

3.1 Some Key Lemmas

For any two integers i', i'' with $i' < i''$, denote by $[i' \ldots i'']$ the sequence of integers between i' and i'', i.e., $[i' \ldots i''] = \{i', i'+1, \ldots, i''\}$. For a functional curve $\mathbf{f}$ defined using the indices in $\{1, 2, \ldots, n\}$, denote by $\mathbf{f}[i' \ldots i'']$ the portion of $\mathbf{f}$ restricted to the indices in $\{i', i'+1, \ldots, i''\}$.

We now give some geometric observations for the unimodal representations. Our purpose is to outline the underlying geometric structures that can be utilized to remodel the UR problem.

Lemma 9 *Let* $\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \ldots, \mathbf{h}^{(k)}$ *be* $k \geq 1$ *unimodal functional curves defined on* $[1 \ldots n]$. *Assume that for each* j, $\mathbf{h}^{(j)}$ *peaks at* i_j^*, *with* $1 \leq i_1^* \leq i_2^* \leq \cdots \leq i_k^* \leq n$. *Then the curve* $\mathbf{h} = \sum_{j=1}^{k} \mathbf{h}^{(j)}$ *satisfies*:

(1) $\mathbf{h}$ *is nonnegative and nondecreasing on* $[1 \ldots i_1^*]$,
(2) $\mathbf{h}$ *is* 0-*UDP-representable on* $[i_j^* \ldots i_{j+1}^*]$ *for each* $j = 1, 2, \ldots, k-1$, *and*
(3) $\mathbf{h}$ *is nonnegative and nonincreasing on* $[i_k^* \ldots n]$.

Proof Since every $\mathbf{h}^{(j)}$ is nondecreasing on $[1 \ldots i_1^*]$ and nonincreasing on $[i_k^* \ldots n]$, (1) and (3) of the lemma follow. (2) of the lemma holds due to the fact that on $[i_j^* \ldots i_{j+1}^*]$, $\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \ldots, \mathbf{h}^{(j)}$ are all nonincreasing, and $\mathbf{h}^{(j+1)}, \mathbf{h}^{(j+2)}, \ldots, \mathbf{h}^{(k)}$ are all nondecreasing. Thus for each j, the portion of the curve $\mathbf{h}$ on $[i_j^* \ldots i_{j+1}^*]$ is equal to the sum of a nondecreasing curve $\mathbf{y}^{(j)} = \sum_{t=j+1}^{k} \mathbf{h}^{(t)}$ and a nonincreasing curve $\mathbf{z}^{(j)} = \sum_{t=1}^{j} \mathbf{h}^{(t)}$. □

Lemma 10 *Given a functional curve* $\mathbf{h}$ *defined on* $[1 \ldots n]$, *if there exist* $k \geq 1$ *integers* $1 \leq i_1^* \leq i_2^* \leq \cdots \leq i_k^* \leq n$ *in* $[1 \ldots n]$ *such that*

(1) $\mathbf{h}$ *is nonnegative and nondecreasing on* $[1 \ldots i_1^*]$,
(2) $\mathbf{h}$ *is* 0-*UDP-representable on* $[i_j^* \ldots i_{j+1}^*]$ *for each* $j = 1, 2, \ldots, k-1$, *and*
(3) $\mathbf{h}$ *is nonnegative and nonincreasing on* $[i_k^* \ldots n]$, *then there exist* k *unimodal curves* $\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \ldots, \mathbf{h}^{(k)}$ *defined on* $[1 \ldots n]$ *such that* $\mathbf{h} = \sum_{j=1}^{k} \mathbf{h}^{(j)}$.

Proof For each $1 \leq j \leq k-1$, since $\mathbf{h}$ is 0-UDP-representable on $[i_j^* \ldots i_{j+1}^*]$, by Theorem 1, we can write $\mathbf{h}[i_j^* \ldots i_{j+1}^*] = \mathbf{y}^{(j)} + \mathbf{z}^{(j)}$, where $\mathbf{y}^{(j)}$ is nondecreasing and $\mathbf{z}^{(j)}$ is nonincreasing on $[i_j^* \ldots i_{j+1}^*]$, with $y_{i_j^*}^{(j)} = 0$ and $z_{i_{j+1}^*}^{(j)} = 0$. If we let $\mathbf{y}^{(0)}$ be $\mathbf{h}[1 \ldots i_1^*]$ and $\mathbf{z}^{(k)}$ be $\mathbf{h}[i_k^* \ldots n]$, then for each $1 \leq j \leq k$, define the unimodal functional curve $\mathbf{h}^{(j)}$ as $\mathbf{y}^{(j-1)} + \mathbf{z}^{(j)}$, more specifically as

$$h_i^{(1)} = \begin{cases} h_i, & i \in [1 \ldots i_1^*], \\ z_i^{(1)}, & i \in [i_1^* \ldots i_2^*], \\ 0, & i \in [i_2^* + 1 \ldots n], \end{cases}$$

$$h_i^{(j)} = \begin{cases} 0, & i \in [1 \ldots i_{j-1}^* - 1], \\ y_i^{(j-1)}, & i \in [i_{j-1}^* \ldots i_j^*], \\ z_i^{(j)}, & i \in [i_j^* \ldots i_{j+1}^*], \\ 0, & i \in [i_{j+1}^* + 1 \ldots n], \end{cases} \qquad \text{for } j = 2, 3, \ldots, k-1,$$

$$h_i^{(k)} = \begin{cases} 0, & i \in [1 \ldots i_{k-1}^* - 1], \\ y_i^{(k)}, & i \in [i_{k-1}^* \ldots i_k^*], \\ h_i, & i \in [i_k^* \ldots n]. \end{cases}$$

Then $\mathbf{h}^{(j)}$ is unimodal on $[1 \ldots n]$ for each $1 \leq j \leq k$, and $\mathbf{h} = \sum_{j=1}^{k} \mathbf{h}^{(j)}$. □

3.2 The min-k Version of the Unimodal Representation Problem

Lemmas 9 and 10 imply that the min-k version of the UR problem on $\mathbf{f}$ and ϵ is equivalent to finding the minimum number of intermediate points $i_1^* \leq i_2^* \leq \cdots \leq i_k^*$ in $[1 \ldots n]$, such that (1) $\mathbf{f}[1 \ldots i_1^*]$ (resp., $\mathbf{f}[i_k^* \ldots n]$) can be represented by a nonnegative nondecreasing (resp., nonincreasing) curve with an error no more than ϵ, (2) for each j with $1 \leq j \leq k-1$, $\mathbf{f}[i_j^* \ldots i_{j+1}^*]$ is ϵ-UDP-representable.

The problem of representing a functional curve by a nonnegative nondecreasing or nonincreasing curve can be solved in a similar spirit as the UDPR feasibility problem, as shown below.

Lemma 11 *Given a nonnegative functional curve* $\mathbf{f} = (f_1, f_2, \ldots, f_n)$ *and* $\epsilon \geq 0$, $\mathbf{f}$ *can be represented by a nonnegative nondecreasing* (*resp*., *nonincreasing*) *curve with*

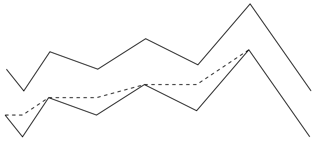

Fig. 11 Illustrating the nondecreasing *curve* (*dashed*) in Lemma 11

an error no bigger than ϵ *if and only if* $f_j - \epsilon \le f_i + \epsilon$ (*resp.*, $f_j + \epsilon \ge f_i - \epsilon$) *holds for all* $1 \le j < i \le n$. *Moreover, if the problem is feasible, then it always has a solution* $\mathbf{y}$ *defined by* $y_i = \max\{0, \max_{j=1}^{i}\{f_j - \epsilon\}\}$ (*resp.*, $y_i = \min_{j=1}^{i}\{f_j + \epsilon\}$), *which can be computed in* $O(n)$ *time* (*see Fig.* 11).

Given $\mathbf{f} = (f_1, f_2, \ldots, f_n)$ and $\epsilon \ge 0$, our min-k algorithm for the UR problem works in a greedy fashion: (1) Find the largest index i_1^* such that $\mathbf{f}[1 \ldots i_1^*]$ can be represented by a nonnegative nondecreasing curve with an error no bigger than ϵ; (2) find the smallest index c, such that $\mathbf{f}$ can be represented by a nonnegative nonincreasing curve on $[c \ldots n]$ with an error no bigger than ϵ; (3) if $i_1^* \ge c$, then we are done; otherwise, by a linear scan from i_1^*, find the largest index i_2^* such that $\mathbf{f}[i_1^* \ldots i_2^*]$ is ϵ-UDP-representable in $O(i_2^* - i_1^*)$ time (by examining each f_i for $i_1^* \le i \le i_2^*$); the same procedure continues until $i_k^* \ge c$. When the algorithm stops, k is the minimum number of unimodal curves needed to represent $\mathbf{f}$ with an error $\le \epsilon$.

In addition to Lemmas 9 and 10, the correctness of the algorithm is also due to the following fact: If $\mathbf{f}$ can be represented by a nonnegative nondecreasing or nonincreasing curve (resp., a pair of nondecreasing and nonincreasing curves) on an interval $[a \ldots b]$ with an error $\le \epsilon$, then f can also be represented by a nonnegative nondecreasing or nonincreasing curve (resp., a pair of nondecreasing and nonincreasing curves) on any sub-interval $[a' \ldots b'] \subseteq [a \ldots b]$ with an error $\le \epsilon$. By Theorem 1 and Lemma 11, the above min-k algorithm takes $O(n)$ time. We thus have the next theorem.

Theorem 3 *The UR* min-k *problem on* $\mathbf{f} = (f_1, f_2, \ldots, f_n)$ *and* $\epsilon \ge 0$ *is solvable in* $O(n)$ *time.*

Additionally, by a somewhat similar proof as that for Lemma 3, we have the following result which will also be useful for our UR min-ϵ algorithm given in the next section.

Lemma 12 *Given a curve* $\mathbf{f} = (f_1, f_2, \ldots, f_n)$ *and* $\epsilon \ge 0$, $\mathbf{f}$ *can be represented by* k *unimodal curves if and only if* SK($\mathbf{f}$) *can be represented by* k *unimodal curves. Furthermore, given a feasible solution for* SK($\mathbf{f}$) (*resp.*, $\mathbf{f}$), *a feasible solution for* $\mathbf{f}$ (*resp.*, SK($\mathbf{f}$)) *can be obtained in* $O(n)$ *time.*

Proof Let f_i be at the point (x_i, f_i) (the ith vertex) of $\mathbf{f}$. The vertices of SK($\mathbf{f}$) are a subset of the vertices of $\mathbf{f}$. Let $|\mathrm{SK}(\mathbf{f})| = m$. For the ith vertex of SK($\mathbf{f}$), $1 \le i \le m$, let $l(i)$ denote the index of the same vertex on $\mathbf{f}$. If $\mathbf{f}$ can be represented by k unimodal curves, then obviously these k unimodal curves can also yield a solution for SK($\mathbf{f}$).

Denote SK($\mathbf{f}$) by $(g_1, g_2, \ldots, g_m)$, with $g_i = f_{l(i)}$ for $1 \le i \le m$. Given $\epsilon \ge 0$, if SK($\mathbf{f}$) can be represented by k unimodal curves $\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \ldots, \mathbf{h}^{(k)}$ (let $\mathbf{h} =$

$\sum_{j=1}^{k}\mathbf{h}^{(j)}=(h_1,h_2,\ldots,h_m)$), then by Lemma 9, we have $1\le i_1^*\le i_2^*\le\cdots\le i_k^*\le m$ such that (1) $g_i-\epsilon\le h_i\le g_i+\epsilon$ for each $1\le i\le m$; (2) $\mathbf{h}$ is nonnegative and nondecreasing on $[1\ldots i_1^*]$; (3) $\mathbf{h}$ is 0-UDP-representable on $[i_j^*\ldots i_{j+1}^*]$ for each $1\le j\le k-1$; (4) $\mathbf{h}$ is nonnegative and nonincreasing on $[i_k^*,m]$. To show that $\mathbf{f}$ can also be represented by k unimodal curves, by Lemma 10, it suffices to find a curve $\mathbf{h}'=(h_1',h_2',\ldots,h_n')$ such that (1) $f_i-\epsilon\le h_i'\le f_i+\epsilon$ for each $1\le i\le n$; (2) $\mathbf{h}'$ is nonnegative and nondecreasing on $[1\ldots l(i_1^*)]$; (3) $\mathbf{h}'$ is 0-UDP-representable on $[l(i_j^*)\ldots l(i_{j+1}^*)]$ for each $1\le j\le k-1$; (4) $\mathbf{h}'$ is nonnegative and nonincreasing on $[l(i_k^*)\ldots n]$.

In the rest of this proof, we denote an index interval $[a\ldots b]$ for $\mathbf{f}$ (resp., $\mathrm{SK}(\mathbf{f})$) by $[a\ldots b]_{\mathbf{f}}$ (resp., $[a\ldots b]_{\mathrm{SK}(\mathbf{f})}$). The curve $\mathbf{h}'$ is constructed from $\mathbf{h}$ in three different ways for the three intervals $[1\ldots l(i_1^*)]_{\mathbf{f}}$, $[l(i_1^*),l(i_k^*)]_{\mathbf{f}}$, and $[l(i_k^*),n]_{\mathbf{f}}$, respectively. For $[1\ldots l(i_1^*)]_{\mathbf{f}}$, for each $i\in[1\ldots i_1^*]_{\mathrm{SK}(\mathbf{f})}$, let $h'_{l(i)}=h_i$; for each $i\in[1\ldots i_1^*]_{\mathrm{SK}(\mathbf{f})}$ and $l(i)<j<l(i+1)$: If $h'_{l(i)}=h'_{l(i+1)}$, then set $h'_j=h'_{l(i)}$; else set $h'_j=f_j+h'_{l(i)}-f_{l(i)}$. It is easy to check that for any $t\in[1\ldots l(i_1^*)]_{\mathbf{f}}$, it must be $f_t-\epsilon\le h'_t\le f_t+\epsilon$ and $\mathbf{h}'$ is nonnegative nondecreasing on $[1\ldots l(i_1^*)]_{\mathbf{f}}$. For $[l(i_1^*),l(i_k^*)]_{\mathbf{f}}$, for each $1\le j<k$, since $\mathbf{h}$ is 0-UDP-representable on $[i_j^*\ldots i_{j+1}^*]_{\mathrm{SK}(\mathbf{f})}$ and $g_t-\epsilon\le h_t\le g_t+\epsilon$ for each $t\in[i_j^*\ldots i_{j+1}^*]_{\mathrm{SK}(\mathbf{f})}$, $\mathrm{SK}(\mathbf{f})$ is ϵ-UDP-representable on $[i_j^*\ldots i_{j+1}^*]_{\mathrm{SK}(\mathbf{f})}$. By Lemma 3, $\mathbf{f}$ is ϵ-UDP-representable on $[l(i_j^*)\ldots l(i_{j+1}^*)]_{\mathbf{f}}$; let its two feasible solution curves be $\mathbf{y}$ and $\mathbf{z}$. Thus the curve $\mathbf{y}+\mathbf{z}$ is 0-UDP-representable on $[l(i_j^*)\ldots l(i_{j+1}^*)]_{\mathbf{f}}$. We let $\mathbf{h}'[l(i_j^*)\ldots l(i_{j+1}^*)]_{\mathbf{f}}$ be $\mathbf{y}+\mathbf{z}$. Thus for each $1\le j\le k-1$, $\mathbf{h}'$ is 0-UDP-representable on $[l(i_j^*)\ldots l(i_{j+1}^*)]_{\mathbf{f}}$ and $f_t-\epsilon\le h'_t\le f_t+\epsilon$ for each $l(i_j^*)\le t\le l(i_{j+1}^*)$. For $[l(i_k^*),n]_{\mathbf{f}}$, we define $\mathbf{h}'$ similarly as for $\mathbf{h}'[1\ldots l(i_1^*)]_{\mathbf{f}}$. The only difference is that $\mathbf{h}'$ is nondecreasing on $[1\ldots l(i_1^*)]_{\mathbf{f}}$, while it is nonincreasing on $[l(i_k^*),n]_{\mathbf{f}}$. □

3.3 The min-ϵ Version of the Unimodal Representation Problem

The UR min-ϵ problem is: Given a functional curve $\mathbf{f}=(f_1,f_2,\ldots,f_n)$ and an integer $k>0$, find the smallest error ϵ^* such that $\mathbf{f}$ can be represented by at most k unimodal curves. In the following, we first give an overview of our algorithm and then discuss the details of the algorithm.

3.3.1 An Overview of the Algorithm

Given a curve $\mathbf{f}$, denote by $\mathcal{K}(\epsilon)$ the minimum number of unimodal curves for representing $\mathbf{f}$ with an error no bigger than ϵ. Clearly, $\mathcal{K}(\epsilon)$ changes in a monotone fashion with respect to ϵ ($\mathcal{K}(\epsilon)$ is a step function like $\mathcal{M}(\epsilon)$ in Fig. 9). To solve the min-ϵ problem, we use our min-k algorithm in the previous section as a black-box search engine, and perform a search for the optimal error ϵ^*. The structures of the unimodal representations specified in Lemmas 9 and 10 imply that we need to consider only those ϵ values that may cause a feasibility change to one of the following representations: (1) representing $\mathbf{f}[i'\ldots i'']$ ($1\le i'<i''\le n$) by a pair of nondecreasing and nonincreasing curves with an error $\le\epsilon$, (2) representing $\mathbf{f}[1\ldots i]$ ($1\le i\le n$) by a nondecreasing curve with an error $\le\epsilon$, or (3) representing $\mathbf{f}[j\ldots n]$ ($1\le j\le n$) by

a nonincreasing curve with an error $\leq \epsilon$. As will be discussed later, the algorithm has two main steps. The first step prunes the errors incurred by the representations of types (2) and (3) above, and the second step prunes the errors incurred by the representations of type (1). Finally, the error ϵ^* is found.

Given a curve $\mathbf{f} = (f_1, f_2, \dots, f_n)$, by Lemma 12, it suffices to consider the UR min-ϵ algorithm for its SK($\mathbf{f}$) curve. After obtaining the minimum error ϵ^* for SK($\mathbf{f}$), we need to use only an additional $O(n)$ time to produce the solution curves for $\mathbf{f}$. The next algorithm focuses on SK($\mathbf{f}$) although it works for any general curve. In the following, we assume SK($\mathbf{f}$) $= \mathbf{g} = (g_1, g_2, \dots, g_m)$ (i.e., $|\mathrm{SK}(\mathbf{f})| = m$).

Given $k > 0$, our UR min-ϵ algorithm has two steps. (1) Search in $S = \{0\} \cup \{|g_i - g_j|/2 \mid 1 \leq i, j \leq m\}$ for $\epsilon', \epsilon'' \in S$, such that ϵ' is the largest element in S with $\mathcal{K}(\epsilon') > k$ and ϵ'' is the smallest element in S with $\mathcal{K}(\epsilon'') \leq k$. (2) With ϵ' and ϵ'', find the smallest value $\epsilon^* \in (\epsilon', \epsilon'']$ with $\mathcal{K}(\epsilon^*) \leq k$. The correctness of this algorithm is obvious.

Note that ϵ' and ϵ'' are two consecutive elements in S in the sense that for any $\hat{\epsilon} \in S$, either $\hat{\epsilon} \leq \epsilon'$ or $\hat{\epsilon} \geq \epsilon''$. Thus, by Lemma 11, changing the error ϵ from ϵ' to ϵ'' (with $\epsilon \in (\epsilon', \epsilon''])$ does not cause a feasibility change on representing $\mathbf{g}[1 \dots i]$ (resp., $\mathbf{g}[j \dots n]$) by an uphill (resp., downhill) curve. Therefore, when ϵ changes from ϵ' to ϵ'', the decreasing of the function $\mathcal{K}(\epsilon)$ is due to the feasibility change of the uphill–downhill pair representations of some $\mathbf{g}[i' \dots i'']$'s, for $1 \leq i' < i'' \leq m$. Denote by $\epsilon[i', i'']$ the minimum error ϵ such that $\mathbf{g}[i' \dots i'']$ is ϵ-UDP-representable, and define $S' = \{\epsilon[i', i''] \mid 1 \leq i' < i'' \leq m\}$. Thus, S' must contain the sought optimal error ϵ^*. The second step is to find ϵ^* in S'. Every step of the algorithm takes $O(m \log m)$ time. The details are given as follows.

3.3.2 The Algorithmic Details

Our algorithm implementation makes use of an interesting technique, which we call *binary search on sorted arrays*, for the following problem: Given M arrays A_i, $1 \leq i \leq M$, each containing $O(N)$ elements in sorted order, find a certain element δ in $A = \bigcup_{i=1}^{M} A_i$. Further, assume that there is a "black-box" decision procedure Π available, such that given any value a, Π reports $a \leq \delta$ or $a > \delta$ in $O(T)$ time. We have the following result.

Lemma 13 *Given M arrays A_i, $1 \leq i \leq M$, each containing $O(N)$ elements in sorted order, a sought element δ in $A = \bigcup_{i=1}^{M} A_i$ can be determined in $O((M+T) \times \log(NM))$ time, where $O(T)$ is the time taken by one call to a "black-box" decision procedure Π.*

Proof The algorithm is of a similar spirit as the linear time selection algorithm [12]. We first sketch the idea and then give the details. For each array A_i, we choose a constant number of its elements as "representative elements". Then, we compute the (weighted) median, say m_a, of these $O(M)$ representative elements, and determine whether $m_a \leq \delta$ by calling the procedure Π, after which half of the representative elements can be pruned. Further, the representative elements are carefully chosen such that a constant fraction of the elements in all M arrays can be pruned. We apply the above procedure recursively on the remaining elements. After $O(\log(NM))$

iterations, the sought element δ is found. In each iteration, we need to compute those $O(M)$ representative elements and their (weighted) median and call the procedure Π once, which altogether take $O(M+T)$ time. The details are given below. Note that the idea above is also somewhat similar to those in [14, 22].

For simplicity of discussion, we assume that every array A_i has at most N elements. Without loss of generality, assume the elements in every array A_i are in nondecreasing order. Below we give the pseudo-code of the algorithm, in which each array A_i maintains a lower index L_i and an upper index U_i such that all currently "active" elements of A_i (i.e., elements that have not been pruned) are between L_i and U_i. Initially, L_i (resp., U_i) is the index of the first (resp., last) element of A_i. Denote the total number of currently "active" elements in A by W. Initially, $W = |A| = O(NM)$.

Pseudo-code

1. If $W \le 7M$, then repeatedly apply the median selection algorithm [12] to the W active elements in A and use binary search to find the sought element δ, where the decision procedure Π is used to determine the search direction. Else, go to Step 2.
2. For each array A_i, if it has less than seven elements (i.e., $U_i - L_i + 1 < 7$), we say that A_i is "not active" and its elements will not be considered until finally in Step 1. Otherwise, it is "active" and we partition its active elements, i.e., those between L_i and U_i, into seven blocks of roughly equal sizes. So every active array is partitioned into seven blocks. Clearly, all active elements of A are partitioned into $O(M)$ blocks.
3. For every such block B_j, let m_j be an arbitrary one of its elements and m_j is considered as a representative element for the block B_j. Let $w_j = |B_j|$ be the weight of m_j. Apply the *weighted* median selection algorithm [12] to find the *weighted* median m' of all $O(M)$ representative elements in $O(M)$ time. Call the procedure Π to determine whether $m' \le \delta$.
4. If $m' \le \delta$, then in each array A_i, let B_j be the block whose median m_j is the largest among all blocks of A_i with $m_j \le m'$ (if there are multiple largest m_j's, take the one with the largest index). Prune all blocks of A_i that are strictly before B_j (i.e., no pruning is done on B_j). Finally, update L_i and U_i for A_i accordingly.
   If $m' > \delta$, the pruning is done in a symmetric way.
5. Using the updated L_i and U_i, recalculate W. Go to Step 1.

The correctness of the above algorithm is obvious from its description and pseudo-code. For the running time, when $W \le 7M$, Step 1 takes $O(M + T\log M)$ time. Recall that T is the time for one call to the procedure Π. In Step 2, every active array is partitioned into seven blocks. It is not difficult to see that this guarantees that the selected weighted median m' prunes away at least one quarter of the active elements in A in Step 4. The running time of Steps 2 to 5 per iteration is clearly $O(M+T)$.

Denote the running time of the algorithm by $T(W)$ (initially $W = O(NM)$). Hence, we have $T(W) \le T(\frac{3W}{4}) + c\cdot(M+T)$ if $W > 7M$ ($c > 0$ is a constant), and $T(W) = O(M + T\log M)$ if $W \le 7M$. Therefore, $T(W) = O((M+T)\log(MN))$. The lemma follows. □

In the following, we discuss our implementation on our UR min-ϵ algorithm, in which Lemma 13 plays an important role. Note that the technique of binary search on sorted arrays discussed above is for solving a very basic problem and we believe our result may find other applications as well.

For Step (1), note that $\mathcal{K}(\epsilon)$ is monotone with respect to ϵ; further, the set S can be represented implicitly as $O(m)$ sorted arrays of size $O(m)$ each. Specifically, after the g_i's are sorted in $O(m\log m)$ time, say, into $g_1, g_2, \dots, g_m$, let array $A_i = \{|g_i - g_j|/2 \mid j = i+1, \dots, m\}$ for every $i = 1, 2, \dots, m-1$. Then, we get m sorted arrays each of which contains at most m elements. Thus, ϵ' and ϵ'' can be found in $O(m\log m)$ time based on Lemma 13, using our UR min-k algorithm as the decision procedure Π.

For Step (2), our task is to find ϵ^* in the set S'. Recall that $S' = \{\epsilon[i', i''] \mid 1 \le i' < i'' \le m\}$, and $\epsilon[i', i'']$ is the minimum error ϵ such that $\mathbf{g}[i' \dots i'']$ is ϵ-UDP-representable. Step (2) can be carried out by performing a similar search as in Step (1) on S' for the optimal value $\epsilon^* \in (\epsilon', \epsilon'']$. Further, since $\epsilon^* \in (\epsilon', \epsilon'']$, we only need to consider those elements of S' which are in $(\epsilon', \epsilon'']$. The key to this hinges on computing efficiently, for any given $1 \le i' < i'' \le m$, the value $\epsilon[i', i'']$ (if it is in $(\epsilon', \epsilon'']$).

We design a data structure such that, after an $O(m)$ time preprocessing, for any query $q(i', i'')$, $1 \le i' < i'' \le m$, the following can be determined in $O(1)$ time: Whether $\epsilon[i', i''] \in (\epsilon', \epsilon'']$; if it is, then report the value of $\epsilon[i', i'']$; otherwise, report whether $\epsilon[i', i''] < \epsilon'$ or $\epsilon[i', i''] > \epsilon''$. Define $\mathcal{G}(\epsilon, \mathbf{g}[i' \dots i''])$ to be $g_{i'} + \epsilon - \mathcal{H}(R(\mathbf{g}[i' \dots i''], \epsilon))$. If we replace $\mathcal{G}(\epsilon)$ in Lemma 5 by $\mathcal{G}(\epsilon, \mathbf{g}[i' \dots i''])$, then by the definition of ϵ' and ϵ'', when $\epsilon \in (\epsilon', \epsilon'']$, $\mathcal{G}(\epsilon, \mathbf{g}[i' \dots i''])$ is a linear function and $\epsilon[i', i'']$ is the unique error $\hat{\epsilon}$ such that $\mathcal{G}(\hat{\epsilon}, \mathbf{g}[i' \dots i'']) = 0$. If $\epsilon[i', i''] \in (\epsilon', \epsilon'']$, then once $\mathcal{G}(\epsilon', \mathbf{g}[i' \dots i''])$ and $\mathcal{G}(\epsilon'', \mathbf{g}[i' \dots i''])$ are available, $\epsilon[i', i'']$ can be obtained in $O(1)$ time. Further, $\epsilon[i', i''] < \epsilon'$ if and only if $\mathcal{G}(\epsilon', \mathbf{g}[i' \dots i'']) > 0$, and $\epsilon[i', i''] > \epsilon''$ if and only if $\mathcal{G}(\epsilon'', \mathbf{g}[i' \dots i'']) < 0$. Thus, to answer each query $q(i', i'')$ in $O(1)$ time, it suffices to compute the two values $\mathcal{H}(R(\mathbf{g}[i' \dots i''], \epsilon'))$ and $\mathcal{H}(R(\mathbf{g}[i' \dots i''], \epsilon''))$ (and consequently, $\mathcal{G}(\epsilon', \mathbf{g}[i' \dots i''])$ and $\mathcal{G}(\epsilon'', \mathbf{g}[i' \dots i''])$) in $O(1)$ time. This is made possible by our $O(m)$ time preprocessing algorithm given below. We only show the preprocessing algorithm for $\mathcal{H}(R(\mathbf{g}[i' \dots i''], \epsilon'))$ (the case for $\mathcal{H}(R(\mathbf{g}[i' \dots i''], \epsilon''))$ is handled similarly).

The main idea for this is to make use of the geometric relations between the (general and long) characteristic curve $R(\mathbf{g}, \epsilon')$ and the (specific and shorter) characteristic curve $R(\mathbf{g}[i' \dots i''], \epsilon')$. More specifically, we use the value $\mathcal{H}(R(\mathbf{g}, \epsilon'))$ to help compute $\mathcal{H}(R(\mathbf{g}[i' \dots i''], \epsilon'))$. As part of the preprocessing, we compute, in $O(m)$ time, the value $\mathcal{H}(R(\mathbf{g}, \epsilon'))$, and further, keep all the *prefix values* $\mathcal{H}(R(\mathbf{g}[1 \dots i], \epsilon'))$ for $1 \le i \le n$. Considering the relations between the characteristic curves $R(\mathbf{g}, \epsilon')$ and $R(\mathbf{g}[i', i''], \epsilon')$, there are two possible cases: (I) The characteristic curve $R(\mathbf{g}[i' \dots i''], \epsilon')$ "merges" into the characteristic curve $R(\mathbf{g}, \epsilon')$ (Fig. 12(a)); (II) $R(\mathbf{g}[i' \dots i''], \epsilon')$ does not merge into $R(\mathbf{g}, \epsilon')$ (Fig. 12(b)). To deal with Case (I), after $R(\mathbf{g}, \epsilon')$ is computed, with an $O(m)$ time preprocessing (given in Lemma 14), we can store the merge point $\overline{i'}$ for every i' in $[1 \dots m]$ (this merge point does not depend on i''), as well as the total amount of "downhill drops" from i' to its $\overline{i'}$ (denote this amount by $C(i')$). In this way, the value of $\mathcal{H}(R(\mathbf{g}[i' \dots i''], \epsilon'))$ is equal to $C(i') + \mathcal{H}(R(\mathbf{g}[1 \dots i''], \epsilon')) - \mathcal{H}(R(\mathbf{g}[1 \dots \overline{i'}], \epsilon'))$ (see Fig. 12(a)), which can be obtained in $O(1)$ time from the prefix values $\mathcal{H}(R(\mathbf{g}[1 \dots i''], \epsilon'))$

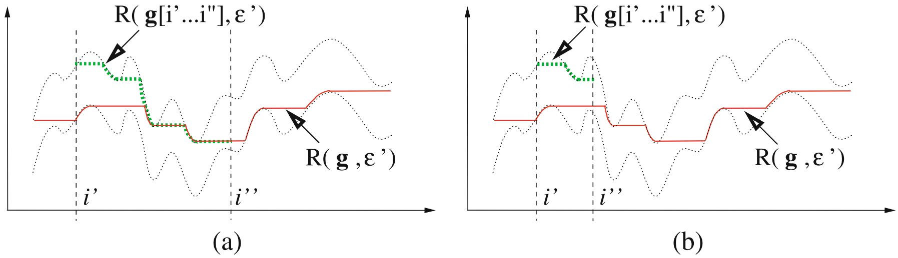

Fig. 12 Illustrating the two cases for computing the representability of $R(\mathbf{g}[i' \dots i''], \epsilon')$: (**a**) $R(\mathbf{g}[i' \dots i''], \epsilon')$ merges into $R(\mathbf{g}, \epsilon')$; (**b**) $R(\mathbf{g}[i' \dots i''], \epsilon')$ does not merge into $R(\mathbf{g}, \epsilon')$

and $\mathcal{H}(R(\mathbf{g}[1 \dots \overline{i'}], \epsilon'))$. Note that the merge point $\overline{i'}$ of i' also allows us to decide in $O(1)$ time which of the two cases holds for a query $q(i', i'')$. For Case (II), the key observation is that the value of $\mathcal{H}(R(\mathbf{g}[i' \dots i''], \epsilon'))$ is $g_{i'} - h[i', i'']$, where $h[i', i'']$ is the minimum value of $\mathbf{g}$ on $[i' \dots i'']$. Thus, with a range minimum data structure [17] (which can be constructed in linear time), we can report $h[i', i'']$, and consequently $\mathcal{H}(R(\mathbf{g}[i' \dots i''], \epsilon'))$, in $O(1)$ time.

Lemma 14 *The merge points* $\bar{i}$*'s for all* $i \in [1 \cdots m]$ *can be obtained in totally* $O(m)$ *time.*

Proof We first compute the two curves $\mathbf{g} + \epsilon'$ and $\mathbf{g} - \epsilon'$, and then the characteristic curve $R(\mathbf{g}, \epsilon')$. In the region $\mathcal{R}$ bounded between $\mathbf{g} + \epsilon'$ and $R(\mathbf{g}, \epsilon')$, we perform a rightwards horizontal trapezoidal decomposition from the vertices of $\mathbf{g} + \epsilon'$. This trapezoidal decomposition can certainly be computed by Chazelle's linear time algorithm in [4], but the problem here is actually much simpler since both $\mathbf{g} + \epsilon'$ and $R(\mathbf{g}, \epsilon')$ are monotone to the x-axis. This produces a set L of horizontal line segments in $\mathcal{R}$. We then connect these segments to $R(\mathbf{g}, \epsilon')$ by following downhill paths along $L \cup (\mathbf{g} + \epsilon')$, until reaching some points on $R(\mathbf{g}, \epsilon')$ (if a point on $R(\mathbf{g}, \epsilon')$ is reachable). Note that for each segment $l \in L$, such a downhill path connecting l to $R(\mathbf{g}, \epsilon')$ (if any) is unique. This process creates a forest, with the whole curve $R(\mathbf{g}, \epsilon')$ being the root of one of the trees, T. For each vertex v of $\mathbf{g} + \epsilon'$ in $\mathcal{R}$, we then find the first point on $R(\mathbf{g}, \epsilon')$ along T, denoted by $\overline{v}$ (if v is not in the tree T containing $R(\mathbf{g}, \epsilon')$, then $\overline{v} = +\infty$). Clearly, these structures can all be built in $O(m)$ time. Thus, in $O(m)$ time, we can compute the merge points for all $i \in [1 \dots m]$. □

Since we are concerned with only those error values in $(\epsilon', \epsilon'']$, for a query $q(i', i'')$, if $\epsilon[i', i''] > \epsilon''$, we simply set $\epsilon[i', i''] = +\infty$. Likewise, if $\epsilon[i', i''] < \epsilon'$, we set $\epsilon[i', i''] = -\infty$. With this value-setting, for any $[j' \dots j''] \subseteq [i' \dots i'']$, we have $\epsilon[j', j''] \leq \epsilon[i', i'']$. Thus, the set S' can be viewed as consisting of $O(m)$ sorted arrays of size $O(m)$ each. Precisely, for each $1 \leq i' \leq m - 1$, let array $A_i = \{\epsilon[i', i''] \mid i'' = i' + 1, \dots, m\}$. Further, S' can be represented implicitly as discussed above and any element of S' can be obtained in $O(1)$ time (after an $O(m)$ time preprocessing). Therefore, by using the searching technique in Lemma 13, we can find the error $\epsilon^* \in S'$ in $O(m \log m)$ time.

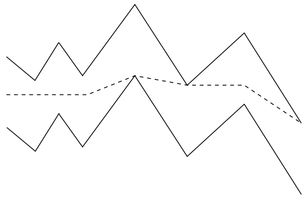

Fig. 13 Illustrating the *curve* $R'(\mathbf{f},\epsilon,\delta)$ (the *dashed one*). The two *solid curves* are $\mathbf{f}+\epsilon$ and $\mathbf{f}-\epsilon$

Theorem 4 *Given an integer* $k>0$, *the UR* min-ϵ *problem on a curve* $\mathbf{f}=(f_1,f_2,\dots,f_n)$ *is solvable in* $O(n+m\log m)$ *time, where* m *is the size of* $\mathrm{SK}(\mathbf{f})$.

4 The Fewer-Peak Representation Problem

In this section, we study the **FPR** problem. Both the min-k and min-ϵ versions are solved by linear time algorithms based on several geometric observations.

4.1 The FPR min-k Algorithm

As in Sect. 2, given a general functional curve $\mathbf{f}$, we first consider the algorithm for its skeleton curve $\mathrm{SK}(\mathbf{f})$, and then handle the original curve $\mathbf{f}$.

Given a skeleton curve $\mathbf{f}=(f_1,f_2,\dots,f_n)$ and $\epsilon\ge 0$, we define a curve $R'(\mathbf{f},\epsilon,\delta)=(R'_1,R'_2,\dots,R'_n)$, where $R'_1=\delta$ and the other $n-1$ curve values are set by using the same rules as the definition of the characteristic curve $R(\mathbf{f},\epsilon)$, namely, for $i=2,3,\dots,n$, set the value of R'_i to $f_i-\epsilon$ if $R'_{i-1}<f_i-\epsilon$, and $f_i+\epsilon$ if $R'_{i-1}>f_i+\epsilon$, and R'_{i-1} otherwise (e.g., see Fig. 13).

Let i' $(1\le i'\le n)$ be the largest integer such that $\bigcap_{i=1}^{i'}[f_i-\epsilon,f_i+\epsilon]=[a,b]$ is not empty. Note that then there exist t and t' with $1\le t\le i'$ and $1\le t'\le i'$ such that $b=f_t+\epsilon$ and $a=f_{t'}-\epsilon$. Our FPR min-k algorithm is based on the following lemma.

Lemma 15 *Given a skeleton curve* $\mathbf{f}=(f_1,f_2,\dots,f_n)$ *and* $\epsilon\ge 0$, $R'(\mathbf{f},\epsilon,\delta)$ *is an optimal solution of the FPR* min-k *problem, where* δ *can be any value in* $[a,b]$ (*see Fig*. 13).

Proof If $i'=n$, then for any $\delta\in[a,b]$, $R'(\mathbf{f},\epsilon,\delta)$ is a horizontal line. Thus $R'(\mathbf{f},\epsilon,\delta)$ has no peak (by our definition of peaks), implying that it is an optimal solution. Below we assume $i'<n$.

According to the definition of i', there are two cases: $f_{i'+1}-\epsilon>b$ or $a>f_{i'+1}+\epsilon$. In the following, we only analyze the former case (the latter case can be handled similarly). Assume that $\mathbf{g}$ is an optimal solution for $\mathbf{f}$ and ϵ. For any $\delta\in[a,b]$, suppose there are k peaks on $R'(\mathbf{f},\epsilon,\delta)$. Below, we prove that $\mathbf{g}$ has at least k peaks, and consequently, $R'(\mathbf{f},\epsilon,\delta)$ is an optimal solution.

Let $\beta(0)=1$. Define $\alpha(1)=\max\{i\mid f_j-\epsilon\le f_i+\epsilon,\text{ for } j=\beta(0),\beta(0)+1,\dots,i\}$ (see Fig. 14). By Lemma 11, $\alpha(1)$ is the rightmost index of $\mathbf{f}$ such that $\mathbf{f}[\beta(0),\alpha(1)]$ (i.e., the portion of $\mathbf{f}$ from $f_{\beta(0)}$ to $f_{\alpha(1)}$) can be represented by

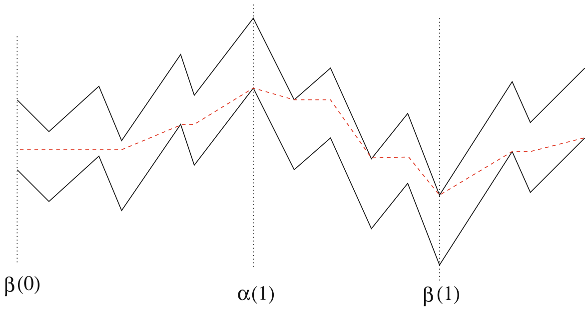

Fig. 14 Illustrating the proof for Lemma 15: The *dashed curve* is $R'(\mathbf{f}, \epsilon, \delta)$

a nonnegative nondecreasing curve with an error at most ϵ. If $\alpha(1) < n$, define $\beta(1) = \max\{i \mid f_j + \epsilon \geq f_i - \epsilon,\ \text{for}\ j = \alpha(1), \alpha(1)+1, \dots, i\}$ (see Fig. 14). By Lemma 11, $\beta(1)$ is the rightmost index of $\mathbf{f}$ such that $\mathbf{f}[\alpha(1), \beta(1)]$ can be represented by a nonnegative nonincreasing curve with an error at most ϵ. If $\alpha(1) = n$, we simply let $\beta(1) = n$. Note that $f_{i'+1} - \epsilon > b$. A simple but critical observation is that $\mathbf{g}$ must have at least one peak in the portion $\mathbf{g}[\beta(0), \beta(1)]$. Further, the curve $R'(\mathbf{f}, \epsilon, \delta)$ has exactly one peak in the portion $R'(\mathbf{f}, \epsilon, \delta)[\beta(0), \beta(1)]$ (see Fig. 14).

If $\beta(1) = n$, then we are done with the proof. Otherwise, similarly, define $\alpha(2) = \max\{i \mid f_j - \epsilon \leq f_i + \epsilon,\ \text{for}\ j = \beta(1), \beta(1)+1, \dots, i\}$. If $\alpha(2) < n$, define $\beta(2) = \max\{i \mid f_j + \epsilon \geq f_i - \epsilon,\ \text{for}\ j = \alpha(2), \alpha(2)+1, \dots, i\}$. If $\alpha(2) = n$, we simply let $\beta(2) = n$. Again, $\mathbf{g}$ must have at least one peak in the portion $\mathbf{g}[\beta(1), \beta(2)]$, and $R'(\mathbf{f}, \epsilon, \delta)$ has exactly one peak in the portion $R'(\mathbf{f}, \epsilon, \delta)[\beta(1), \beta(2)]$. If we repeat the same procedure as above until either $\alpha(i)$ or $\beta(i)$ is n, we can obtain that the number of peaks in $\mathbf{g}$ is at least k, i.e., the number of peaks in $R'(\mathbf{f}, \epsilon, \delta)$. □

It is easy to check that for any $\epsilon > 0$, for any two values δ_1 and δ_2 both in $[a, b]$, R'_i is a peak on $R'(\mathbf{f}, \epsilon, \delta_1)$ if and only if R'_i is a peak on $R'(\mathbf{f}, \epsilon, \delta_2)$. Since $R'(\mathbf{f}, \epsilon, \delta)$ can be easily computed in linear time, the FPR min-k problem on a skeleton curve is solvable in linear time. By following a similar proof as for Lemma 3, we obtain the following lemma.

Lemma 16 *For any general functional curve* $\mathbf{f}$ *and* $\epsilon \geq 0$, $\mathbf{f}$ *can be approximated by a k-peak function if and only if* $\mathrm{SK}(\mathbf{f})$ *can be approximated by a k-peak function. Furthermore, given a feasible solution for* $\mathrm{SK}(\mathbf{f})$ (*resp.*, $\mathbf{f}$), *a feasible solution for* $\mathbf{f}$ (*resp.*, $\mathrm{SK}(\mathbf{f})$) *can be obtained in $O(n)$ time.*

Proof The proof is quite similar to that for Lemma 3. We only sketch it here. Let f_i be at the point (x_i, f_i) (the ith vertex) of $\mathbf{f}$. Let $|\mathrm{SK}(\mathbf{f})| = m$. For the ith vertex of $\mathrm{SK}(\mathbf{f})$, $1 \leq i \leq m$, let $l(i)$ denote the index of the same vertex on $\mathbf{f}$. If $\mathbf{f}$ can be represented by a k-peak function, then obviously $\mathrm{SK}(\mathbf{f})$ can also be represented by the same k-peak function.

Given $\mathbf{f} = (f_1, f_2, \dots, f_n)$ and $\epsilon \geq 0$, suppose by Lemma 15 we find a k-peak function $\mathbf{g} = (g_1, g_2, \dots, g_m)$ to represent $\mathrm{SK}(\mathbf{f})$ with an error no bigger than ϵ. Let $\mathbf{g}$'s profile curves be a nondecreasing curve $\mathbf{y} = (y_1, y_2, \dots, y_m)$ and a nonincreasing curve $\mathbf{z} = (z_1, z_2, \dots, z_m)$. In the same way as for the proof of Lemma 3, we can find a nondecreasing curve $\mathbf{y}' = (y'_1, y'_2, \ldots, y'_n)$ and a nonincreasing curve $\mathbf{z}' = (z'_1, z'_2, \ldots, z'_n)$ such that (1) $f_i - \epsilon \le y'_i + z'_i \le f_i + \epsilon$ for each $1 \le i \le n$, and (2) if $\mathbf{g}$ is nonincreasing (resp., nondecreasing) from g_i to g_{i+1}, then $\mathbf{g}' = \mathbf{y}' + \mathbf{z}'$ is also nonincreasing (resp., nondecreasing) from $g'_{l(i)}$ to $g'_{l(i+1)}$. It is easy to see that the curve $\mathbf{g}'$ thus obtained is a feasible k-peak function for $\mathbf{f}$ and ϵ, and can be computed in $O(n)$ time. □

Theorem 5 *The FPR* min-k *problem on* $\mathbf{f} = (f_1, f_2, \ldots, f_n)$ *and* $\epsilon \ge 0$ *is solvable in* $O(n)$ *time.*

4.2 The FPR min-ϵ Algorithm

Given a skeleton curve $\mathbf{f}$, when $\epsilon = 0$, we have $\delta = a = b = f_1$ and $R'(\mathbf{f}, \epsilon, \delta)$ is exactly the curve $\mathbf{f}$. Intuitively, as ϵ increases from 0 to ∞, the number of peaks on $R'(\mathbf{f}, \epsilon, \delta)$ decreases, that is, some peaks on $R'(\mathbf{f}, \epsilon, \delta)$ disappear. If we define the R-peaks, $P(\mathbf{f})$, and ϵ'_i (for each $i \in P(\mathbf{f})$) in the same way as in Lemma 6, then by following a similar proof as for Lemma 6, we have the next lemma.

Lemma 17 *For any* $\epsilon \ge 0$, *if* $\delta \in [a, b]$, *then for each* $1 < i < n$, R'_i *is an R-peak on* $R'(\mathbf{f}, \epsilon, \delta)$ *if and only if* $i \in P(\mathbf{f})$ *and* $\epsilon < \epsilon'_i$.

Geometrically, when ϵ increases from 0 to ∞, ϵ^* is the minimum error when there are at most k peaks left on $R'(\mathbf{f}, \epsilon, \delta)$. Define the multi-set E as $\{\epsilon'_i \mid i \in P(\mathbf{f})\}$. Since the peaks on $R'(\mathbf{f}, \epsilon, \delta)$ consist of all its R-peaks and possibly R'_1 and R'_n if one of them or both are peaks, by Lemma 17, ϵ^* must be one of the $(m-k)$th, $(m-k+1)$th, and $(m-k+2)$th smallest elements in E, where m is the number of peaks on $\mathbf{f}$. Thus, ϵ^* can be obtained by the following theorem.

Theorem 6 *The FPR* min-ϵ *problem on* $\mathbf{f} = (f_1, f_2, \ldots, f_n)$ *and* $k \ge 0$ *is solvable in* $O(n)$ *time.*

Proof A straightforward linear time algorithm works as follows. (1) Compute the $(m-k)$th smallest element in E and let it be ϵ'. (2) Compute $R'(\mathbf{f}, \epsilon', \delta)$, and if there are no more than k peaks, then $\epsilon^* = \epsilon'$; otherwise, go to the next step. (3) Compute the $(m-k+1)$th smallest element in E and let it be ϵ''. (4) Compute $R'(\mathbf{f}, \epsilon'', \delta)$, and if there are no more than k peaks, then $\epsilon^* = \epsilon''$; otherwise, ϵ^* is the $(m-k+2)$th smallest element in E.

Note that the above algorithm assumes $k \ge 2$. When $k = 0$ or $k = 1$, the problem can be solved in linear time in a similar way. □

Acknowledgement The authors would like to thank an anonymous referee for several useful suggestions that helped improve the presentation of the paper.

References

1. Bansal, N., Coppersmith, D., Schieber, B.: Minimizing setup and beam-on times in radiation therapy. In: Proc. of the 9th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, pp. 27–38 (2006)

2. Barequet, G., Chen, D.Z., Daescu, O., Goodrich, M., Snoeyink, J.: Efficiently approximating polygonal paths in three and higher dimensions. Algorithmica **33**(2), 150–167 (2002)
3. Chan, S., Chin, F.: Approximation of polygonal curves with minimum number of line segments or minimum error. Int. J. Comput. Geom. Appl. **6**, 59–77 (1996)
4. Chazelle, B.: Triangulating a simple polygon in linear time. Discrete Comput. Geom. **6**, 485–524 (1991)
5. Chen, D.Z., Daescu, O.: Space-efficient algorithms for approximating polygonal curves in two dimensional space. Int. J. Comput. Geom. Appl. **13**(2), 95–111 (2003)
6. Chen, D.Z., Chun, J., Katoh, N., Tokuyama, T.: Efficient algorithms for approximating a multidimensional voxel terrain by a unimodal terrain. In: Proc. of the 10th Annual International Computing and Combinatorics Conference, pp. 238–248 (2004)
7. Chen, D.Z., Hu, X.S., Luan, S., Misiolek, E., Wang, C.: Shape rectangularization problems in intensity-modulated radiation therapy. In: Proc. of the 12th Annual Int. Symp. on Algorithms and Computation, pp. 701–711 (2006)
8. Chen, D.Z., Hu, X.S., Luan, S., Naqvi, S.A., Wang, C., Yu, C.X.: Generalized geometric approaches for leaf sequencing problems in radiation therapy. Int. J. Comput. Geom. Appl. **16**(2–3), 175–204 (2006)
9. Chun, J., Sadakane, K., Tokuyama, T., Yuki, M.: Peak-reducing fitting of a curve under the L_p metric. Interdiscip. Inf. Sci. **11**(2), 191–198 (2005)
10. Chun, J., Sadakane, K., Tokuyama, T.: Efficient algorithms for constructing a pyramid from a terrain. IEICE Trans. Inf. Syst. **E89-D**(2), 783–788 (2006)
11. Chun, J., Sadakane, K., Tokuyama, T.: Linear time algorithm for approximating a curve by a single-peaked curve. Algorithmica **44**(2), 103–115 (2006)
12. Cormen, T., Leiserson, C., Rivest, R., Stein, C.: Introduction to Algorithms, 2nd edn. MIT Press, Cambridge (2001)
13. Edelsbrunner, H., Harer, J.: Persistent homology—a survey. In: Goodman, J.E., Pach, J., Pollack, R. (eds.) Twenty Years After. AMS, Providence (2007)
14. Frederickson, G., Johnson, D.: The complexity of selection and ranking in $X + Y$ and matrices with sorted columns. J. Comput. Syst. Sci. **24**(2), 197–208 (1982)
15. Fukuda, T., Morimoto, Y., Morishita, S., Tokuyama, T.: Implementation and evaluation of decision trees with range and region splitting. Constraints **2**, 401–427 (1997)
16. Fukuda, T., Morimoto, Y., Morishita, S., Tokuyama, T.: Data mining with optimized two-dimensional association. ACM Trans. Database Syst. **26**, 179–213 (2001)
17. Gabow, H., Bentley, J., Tarjan, R.: Scaling and related techniques for geometry problems. In: Proc. of the 16th Annual ACM Symposium on Theory of Computing (STOC), pp. 135–143 (1984)
18. Goodrich, M.: Efficient piecewise-linear function approximation using the uniform metric. In: Proc. of the 10th Annual ACM Symposium on Computational Geometry, pp. 322–331 (1994)
19. Guha, S., Shim, K.: A note on linear time algorithms for maximum error histograms. IEEE Trans. Knowl. Data Eng. **19**(7), 993–997 (2007)
20. Hardwick, J., Stout, Q.F.: Optimizing a unimodal response function for binary variables. In: Atkinson, A., Bogacka, B., Zhigljavsky, A. (eds.) Optimum Design, pp. 195–208 (2000)
21. Imai, H., Iri, M.: Computational-geometric methods for polygonal approximations of a curve. Comput. Vis. Graph. Image Process. **36**, 31–41 (1986)
22. Krizanc, D., Morin, P., Smid, M.: Range mode and range median queries on lists and trees. Nord. J. Comput. **12**(1), 1–17 (2005)
23. O'Rourke, J.: An on-line algorithm for fitting straight lines between data ranges. Commun. ACM **24**, 574–578 (1981)
24. Pan, G.: Subset selection with additional order information. Biometrics **52**(4), 1363–1374 (1996)
25. Stout, Q.F.: Unimodal regression via prefix isotonic regression. Comput. Stat. Data Anal. **53**(2), 289–297 (2008)
26. Turner, T.R., Wollan, P.C.: Locating a maximum using isotonic regression. Comput. Stat. Data Anal. **25**(3), 305–320 (1997)
27. Varadarajan, K.: Approximating monotone polygonal curves using the uniform metric. In: Proc. of the 12th Annual ACM Symposium on Computational Geometry, pp. 311–318 (1996)
28. Wang, D., Huang, N., Chao, H., Lee, R.: Plane sweep algorithms for the polynomial approximation problems with applications. In: Proc. of the 4th International Symposium on Algorithms and Computation (ISAAC), pp. 515–522 (1993)

29. Wang, C., Luan, S., Tang, G., Chen, D.Z., Earl, M.A., Yu, C.X.: Arc-modulated radiation therapy (AMRT): A single-arc form of intensity-modulated arc therapy. Phys. Med. Biol. **53**(22), 6291–6303 (2008)
30. Webb, S.: The Physics of Three-Dimensional Radiation Therapy. Institute of Physics, Bristol (1993)
31. Webb, S.: The Physics of Conformal Radiotherapy—Advances in Technology. Institute of Physics, Bristol (1997)
32. Zomorodian, A., Carlsson, G.: Computing persistent homology. Discrete Comput. Geom. **33**(2), 249–274 (2005)