Rollercoasters: Long Sequences without Short Runs

Therese Biedl[1], Ahmad Biniaz[1], Robert Cummings[1], Anna Lubiw[1], Florin Manea[2], Dirk Nowotka[2], and Jeffrey Shallit[1]

1 **Cheriton School of Computer Science, University of Waterloo, Canada.**
2 **Department of Computer Science, Kiel University, Germany.**

Abstract

A *rollercoaster* is a sequence of real numbers for which every maximal contiguous subsequence, that is increasing or decreasing, has length at least three. By translating this sequence to a set of points in the plane, a rollercoaster can be defined as a polygonal path for which every maximal sub-path, with positive- or negative-slope edges, has at least three points. Given a sequence of distinct real numbers, the rollercoaster problem asks for a maximum-length (not necessarily contiguous) subsequence that is a rollercoaster. It was conjectured that every sequence of n distinct real numbers contains a rollercoaster of length at least $\lceil n/2 \rceil$ for $n > 7$, while the best known lower bound is $\Omega(n/\log n)$. In this paper we prove this conjecture. Our proof is constructive and implies a linear-time algorithm for computing a rollercoaster of this length. Extending the $O(n \log n)$-time algorithm for computing a longest increasing subsequence, we show how to compute a maximum-length rollercoaster within the same time bound. A maximum-length rollercoaster in a permutation of $\{1, \dots, n\}$ can be computed in $O(n \log \log n)$ time.

The search for rollercoasters was motivated by orthogeodesic point-set embedding of caterpillars. A *caterpillar* is a tree such that deleting the leaves gives a path, called the *spine*. A *top-view caterpillar* is one of degree 4 such that the two leaves adjacent to each vertex lie on opposite sides of the spine. As an application of our result on rollercoasters, we are able to find a planar drawing of every n-node top-view caterpillar on every set of $\frac{25}{3}n$ points in the plane, such that each edge is an orthogonal path with one bend. This improves the previous best known upper bound on the number of required points, which is $O(n \log n)$. We also show that such a drawing can be obtained in linear time, provided that the points are given in sorted order.

1 Introduction

A *run* in a sequence of real numbers is a maximal contiguous subsequence that is increasing (an "ascent") or decreasing (a "descent"). A *rollercoaster* is a sequence of real numbers such that every run has length at least three. For example the sequence $(8, 5, 1, 3, 4, 7, 6, 2)$ is a rollercoaster with runs $(8, 5, 1)$, $(1, 3, 4, 7)$, $(7, 6, 2)$, which have lengths 3, 4, 3, respectively. The sequence $(8, 5, 1, 7, 6, 2, 3, 4)$ is not a rollercoaster because its run $(1, 7)$ has length 2. Given a sequence $S = (s_1, s_2, \dots, s_n)$ of n distinct real numbers, the rollercoaster problem is to find a maximum-size set of indices $i_1 < i_2 < \dots < i_k$ such that $(s_{i_1}, s_{i_2}, \dots, s_{i_k})$ is a rollercoaster. In other words, this problem asks for a longest rollercoaster in S, i.e., a longest subsequence of S that is a rollercoaster.

One can interpret S as a set P of points in the plane by translating each number $s_i \in S$ to a point $p_i = (i, s_i)$. With this translation, a rollercoaster in S translates to a "rollercoaster" in P, which is a polygonal path whose vertices are points of P and such that every maximal sub-path, with positive- or negative-slope edges, has at least three points. See Figure 1(a). Conversely, for any point set in the plane, the y-coordinates of the points, ordered by their x-coordinates, forms a sequence of numbers. Therefore, any rollercoaster in P translates to a rollercoaster of the same length in S.

34th European Workshop on Computational Geometry, Berlin, Germany, March 21–23, 2018.

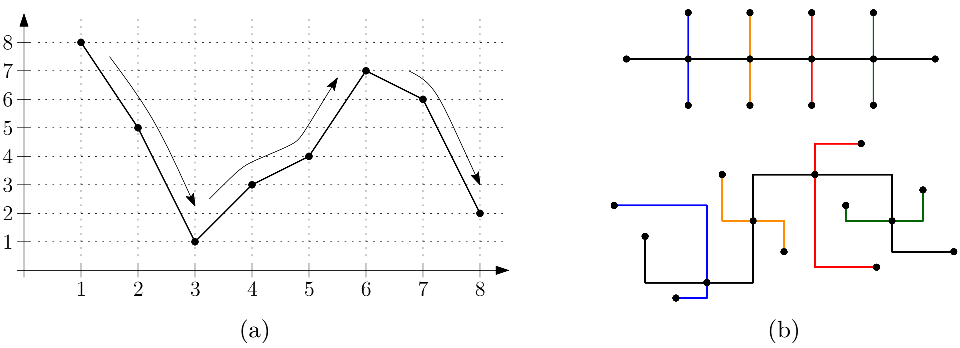

Figure 1 (a) Translating the sequence $(8, 5, 1, 3, 4, 7, 6, 2)$ to a set of points. (b) A planar L-shaped drawing of a top-view caterpillar.

The best known lower bound on the length of a longest rollercoaster is $\Omega(n/\log n)$ due to Biedl et al. [2]. They conjectured that

▶ Conjecture 1.1. Every sequence of $n > 7$ distinct real numbers contains a rollercoaster of length at least $\lceil n/2 \rceil$.

Conjecture 1.1 can be viewed as a statement about patterns in permutations, a topic with a long history, and the subject of much current research. For example, the Eulerian polynomials, introduced by Euler in 1749, are the generating function for the number of descents in permutations. For surveys of recent work, see, for example, Linton et al. [7] and Kitaev [6]. Specifically, Conjecture 1.1 is related to the following seminal result of Erdős and Szekeres [3] in the sense that they prove the existence of an increasing or a decreasing subsequence of length at least $\sqrt{n}+1$ for $n = ab + 1$, which is essentially a rollercoaster with one run.

▶ **Theorem 1.2** (Erdős and Szekeres, 1935). *Every sequence of $ab + 1$ distinct real numbers contains an increasing subsequence of length at least $a + 1$ or a decreasing subsequence of length at least $b + 1$.*

Hammersley [5] gave an elegant proof of the Erdős-Szekeres theorem that is short, simple, and based on the pigeonhole principle. The Erdős-Szekeres theorem also follows from the well-known decomposition of Dilworth (see [9]). The following is a restatement of Dilworth's decomposition for sequences of numbers.

▶ **Theorem 1.3** (Dilworth, 1950). *Any finite sequence S of distinct real numbers can be partitioned into k ascending sequences where k is the maximum length of a descending sequence in S.*

Besides its inherent interest, the study of rollercoasters is motivated by point-set embedding of caterpillars [2]. A *caterpillar* is a tree such that deleting the leaves gives a path, called the *spine*. An *ordered caterpillar* is a caterpillar in which the cyclic order of edges incident to each vertex is specified. A *top-view caterpillar* is an ordered caterpillar where all vertices have degree 4 or 1 such that the two leaves adjacent to each vertex lie on opposite sides of the spine. Planar orthogonal drawings of trees on a fixed set of points in the plane have been explored recently, see e.g., [2, 4, 8]; in these drawings every edge is drawn as an orthogonal path between two points, and the edges are non-intersecting. A *planar L-shaped drawing* is a simple type of planar orthogonal drawing in which every edge is an orthogonal path of exactly two segments. Such a path is called an *L-shaped edge*. For example see the top-view caterpillar in Figure 1(b) together with a planar L-shaped drawing on a given point set. Biedl et al. [2] proved that every top-view caterpillar on n vertices has a planar L-shaped drawing on every set of $O(n \log n)$ points in the plane that is in *general orthogonal position*, meaning that no two points have the same x- or y-coordinate.

Due to space restrictions we cannot give all the proofs. We refer the interested reader to the full version [1].

2 Rollercoasters

Our main result is to show that Conjecture 1.1 holds. In fact we prove something stronger: every sequence of n distinct numbers contains two rollercoasters of total length n. Our proof is constructive and yields a linear-time algorithm for computing such rollercoasters. The length 4 sequence $(3, 4, 1, 2)$ has no rollercoaster, and it can be shown that for $n = 5, 6, 7$ the longest rollercoaster has length 3. Therefore, we only consider $n \geq 8$.

▶ **Theorem 2.1.** *Every sequence of $n \geqslant 8$ distinct real numbers contains a rollercoaster of length at least $\lceil n/2 \rceil$; such a rollercoaster can be computed in linear time. The lower bound of $\lceil n/2 \rceil$ is tight in the worst case.*

Proof. Consider a sequence with $n \geqslant 8$ distinct real numbers, and let P be its point-set translation with points $p_1, \dots, p_n$ that are ordered from left to right. We define a *pseudo-rollercoaster* as a sequence in which every run is a 3-ascent (an ascent of length at least 3) or a 3-descent, except possibly the first run. We present an algorithm that computes two pseudo-rollercoasters R_1 and R_2 in P such that $|R_1| + |R_2| \geqslant n$; the length of the longer one is at least $\lceil n/2 \rceil$. Then with a more involved proof we show how to extend this longer pseudo-rollercoaster to obtain a rollercoaster of length at least $\lceil n/2 \rceil$; this will prove the lower bound.

First we provide a high-level description of our algorithm as depicted in Figure 2. Our algorithm is iterative, and proceeds by sweeping the plane by a vertical line ℓ from left to right. We maintain the following invariant: At the beginning of every iteration we have two pseudo-rollercoasters whose union is the set of all points to the left of ℓ and such that the last run of one of them is an ascent and the last run of the other one is a descent. Furthermore, these two last runs have a point in common.

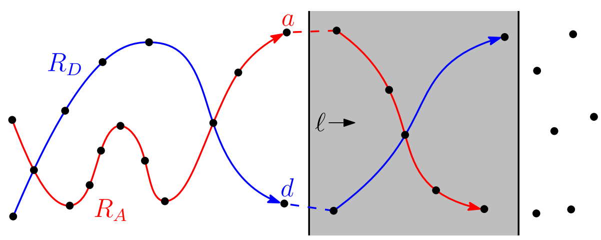

Figure 2 One iteration of algorithm: Constructing two pseudo-rollercoasters.

During every iteration we move ℓ forward and try to extend the current pseudo-rollercoasters. If this is not immediately possible with the next point, then we move ℓ farther and stop as soon as we are able to split all the new points into two chains that can be appended to the current pseudo-rollercoasters to obtain two new pseudo-rollercoasters that satisfy the invariant. See Figure 2. Now we present our iterative algorithm in detail.

The First Iteration: We take the leftmost point p_1, and initialize each of the two pseudo-rollercoasters by p_1 alone. We may consider one of the pseudo-rollercoasters to end in an ascent and the other pseudo-rollercoaster to end in a descent. The two runs have a point in common.

An Intermediate Iteration: By the above invariant we have two pseudo-rollercoasters R_A and R_D whose union is the set of all points to the left of ℓ and such that the last run of one of them, say R_A, is an ascent and the last run of R_D is a descent. Furthermore, the last run of R_A and the last run of R_D have a point in common. During the current iteration we make sure that every swept point will be added to R_A or R_D or both. We also make sure that at the end of this iteration the invariant will hold for the next iteration. Let a and d denote the rightmost points of R_A and R_D, respectively; see Figure 2. Let p_i be the first point to the right of ℓ. If p_i is above a, we add p_i to R_A to complete this iteration. Similarly, if p_i is below d, we add p_i to R_D to complete this iteration. In either case we get two pseudo-rollercoasters that satisfy the invariant for the next iteration. Thus we may assume that p_i lies below a and above d. In particular, this means that a lies above d.

Consider the next point p_{i+1}. (If there is no such point, go to the last iteration.) Suppose without loss of generality that p_{i+1} lies above p_i as depicted in Figure 3. Then d, p_i, p_{i+1} forms a 3-ascent. Continue considering points $p_{i+2}, \dots, p_k$ until for the first time, there is a 3-descent in $a, p_i, \dots, p_k$. In other words, k is the smallest index for which $a, p_i, \dots, p_k$ contains a descending chain of length 3. (If we run out of points before finding a 3-descent, then go to the last iteration.)

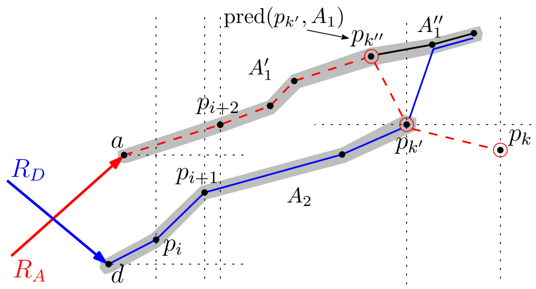

Figure 3 Illustration of an intermediate iteration of the algorithm.

Without p_k there is no descending chain of length 3. Thus the longest descending chain has two points, and by Theorem 1.3, the sequence $P' = a, p_i, p_{i+1}, \dots, p_{k-1}$ is the union of two ascending chains. We give an algorithm to find two such chains A_1 and A_2 with A_1 starting at a and A_2 starting at p_i. The algorithm also finds the 3-descent ending with p_k. For every point $q \in A_2$ we define its A_1-*predecessor* to be the rightmost point of A_1 that is to the left of q. We denote the A_1-predecessor of q by pred(q, A_1).

The algorithm is as follows: While moving ℓ forward, we denote by r_1 and r_2 the rightmost points of A_1 and A_2, respectively; at the beginning $r_1 = a$, $r_2 = p_i$, and pred$(p_i, A_1) = a$. Let p be the next point to be considered. If p is above r_1 then we add p to A_1. If p is below r_1 and above r_2, then we add p to A_2 and set pred$(p, A_1) = r_1$. If p is below r_2, then we find our desired first 3-descent formed by (in backwards order) $p_k = p$, $p_{k'} = r_2$, and $p_{k''} =$ pred(r_2, A_1). See Figure 3. This algorithm runs in time $O(k-i)$, which is proportional to the number of swept points.

We add point d to the start of chain A_2. The resulting chains A_1 and A_2 are shaded in Figure 3. Observe that A_2 ends at $p_{k'}$. Also, all points of P' that are to the right of $p_{k'}$ (if there are any) belong to A_1, and lie to the right of $p_{k''}$, and form an ascending chain. Let A_1'' be this ascending chain. Let A_1' be the sub-chain of A_1 up to $p_{k''}$; see Figure 3. Now we form one pseudo-rollercoaster (shown in red/dashed) consisting of R_A followed by A_1' and then by the descending chain $p_{k''}, p_{k'}, p_k$. We form another pseudo-rollercoaster (shown in blue/solid) consisting of R_D followed by A_2 and then by A_1''. We need to verify that the ascending chain added after d has length at least 3. This chain contains d, p_i and $p_{k'}$. This gives a chain of length at least 3 unless $k' = i$, but in this case $p_{k''} = a$, so p_{i+1} is part of A_1'' and consequently part of this ascending chain. Thus we have constructed two longer pseudo-rollercoasters whose union is the set of all points up to point p_k, one ending with a 3-ascent and one with a 3-descent and such that the last two runs share the point $p_{k'}$. Figure 4(a) shows an intermediate iteration.

The Last Iteration: If there are no points left, then we terminate the algorithm. Otherwise, let p_i be the first point to the right of ℓ. Let a and d be the endpoints of the two pseudo-rollercoasters obtained so far, such that a is the endpoint of an ascent and d is the endpoint of a descent. Notice that p_i is below a and above d, because otherwise this iteration would be an intermediate one. For the same reason, the remaining points $p_i, \dots, p_n$ do not contain a 3-ascent together with a 3-descent. If p_i is the last point, i.e., $i = n$, then we discard this point and terminate this iteration. Assume that $i \neq n$, and suppose without loss of generality that the next point p_{i+1} lies above p_i. In this setting, by Theorem 1.3 and as described in an intermediate iteration, with the remaining points, we can get two ascending chains A_1 and A_2 such that A_2 contains at least two points. By connecting A_1 to a and A_2 to d we get two pseudo-rollercoasters whose union is all the points (in this iteration we do not need to maintain the invariant).

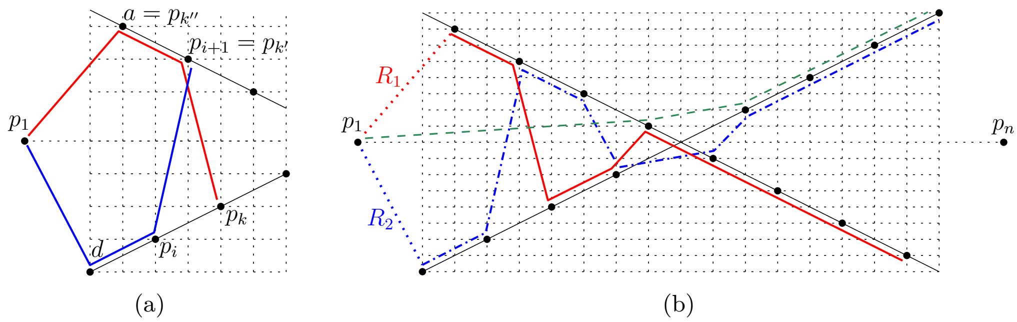

Figure 4 (a) An intermediate iteration. (b) A point set for which any rollercoaster of length at least $n/4 + 3$ does not contain p_1 and p_n. The green (dashed) rollercoaster, which contains p_1, has length $n/4 + 2$. The red (solid) and blue (dash-dotted) chains are the two rollercoasters returned by our algorithm.

Final Refinement: At the end of the algorithm, we obtain two pseudo-rollercoasters R_1 and R_2 that share p_1, and their union contains all points of P, except possibly p_n. Thus, $|R_1| + |R_2| \geqslant n$, and the length of the longer one is at least $\lceil \frac{n}{2} \rceil$.

This ends the presentation of our algorithm. It is not hard to see that the algorithm runs in $O(n)$ time.

To obtain rollercoasters (not just pseudo-rollercoasters), we remove p_1 from R_1 and/or R_2 if the first run only contains two points. This gives two rollercoasters $\mathcal{R}_1$ and $\mathcal{R}_2$ whose union contains all points, except possibly p_1 and p_n. The length of the longer one is at least $\lceil \frac{n-2}{2} \rceil$. We can improve this bound to $\lceil \frac{n}{2} \rceil$ by revisiting the first and last iterations of our algorithm with some case analysis.

We note that there are point sets, with n points, for which every rollercoaster of length at least $n/4 + 3$ does not contain any of p_1 and p_n; see e.g., the point set in Figure 4(b). To verify the tightness of the $\lceil n/2 \rceil$ lower bound, consider a set of n points in the plane where

$\lceil n/2 \rceil$ of which lie on a positive-slope line segment in the $(-,+)$-quadrant and the other $\lfloor n/2 \rfloor$ points lie on a positive-slope lines segment in the $(+,-)$-quadrant. ◀

3 Further Results

Our result can be extended to k-rollercoasters, i.e., sequences of real numbers in which every run is either a k-ascent or a k-descent. Namely, for $k \geqslant 4$, every sequence of $n \geqslant (k-1)^2+1$ distinct real numbers contains a k-rollercoaster of length at least $\frac{n}{2(k-1)} - \frac{3k}{2}$.

The algorithm presented in the proof of Theorem 2.1 does not necessarily compute the longest rollercoaster in a sequence. This can be done in $O(n \log n)$-time by an algorithm extending the classical algorithm for computing a longest increasing subsequence. This algorithm can be implemented in $O(n \log\log n)$ time if each number in the input sequence is an integer that fits in a constant number of memory words. Connected to this last result, we give an estimate on the number of permutations of $\{1,\ldots,n\}$ that are rollercoasters. Namely, let $r(n)$ be the number of permutations of $\{1,2,\ldots,n\}$ that are rollercoasters. We show that $r(n) \sim c' \cdot n! \cdot \lambda^{n-3}$ where c' is a constant, approximately 0.204.

Finally, we study the problem of drawing a top-view caterpillar, with L-shaped edges, on a set of points in the plane that is in general orthogonal position. Recall that a top-view caterpillar is an ordered caterpillar of degree 4 such that the two leaves adjacent to each vertex lie on opposite sides of the spine; see Figure 1(b) for an example. The best known upper bound on the number of required points for a planar L-shaped drawing of every n-vertex top-view caterpillar is $O(n \log n)$; this bound is due to Biedl et al. [2]. We use Theorem 2.1 and improve this bound to $\frac{25}{3}n+O(1)$.

▶ **Theorem 3.1.** *Any top-view caterpillar of n vertices has a planar L-shaped drawing on any set of $\frac{25}{3}n + O(1)$ points in the plane that is in general orthogonal position.*

References

1 Therese Biedl, Ahmad Biniaz, Robert Cummings, Anna Lubiw, Florin Manea, Dirk Nowotka, and Jeffrey Shallit. Rollercoasters and caterpillars. *CoRR*, abs/1801.08565, 2018.

2 Therese Biedl, Timothy M. Chan, Martin Derka, Kshitij Jain, and Anna Lubiw. Improved bounds for drawing trees on fixed points with L-shaped edges. *CoRR*, abs/1709.01456, 2017. Also to appear in *Proceedings of GD, 2017*.

3 Paul Erdős and George Szekeres. A combinatorial problem in geometry. *Compositio Mathematica*, 2:463–470, 1935.

4 Emilio Di Giacomo, Fabrizio Frati, Radoslav Fulek, Luca Grilli, and Marcus Krug. Orthogeodesic point-set embedding of trees. *Computational Geometry: Theory and Applications*, 46(8):929–944, 2013.

5 John M. Hammersley. A few seedlings of research. In *Proceedings of the 6th Berkeley Symposium on Mathematical Statistics and Probability*, pages 345–394. University of California Press, 1972.

6 Sergey Kitaev. *Patterns in Permutations and Words*. Springer, 2011.

7 S. Linton, N. Ruškuc, and V. Vatter, editors. *Permutation Patterns*. London Mathematical Society Lecture Note Series, vol. 376, Cambridge, 2010.

8 Manfred Scheucher. Orthogeodesic point set embeddings of outerplanar graphs. Master's thesis, Graz University of Technology, 2015.

9 J. Michael Steele. Variations on the monotone subsequence theme of Erdős and Szekeres. In David Aldous, Persi Diaconis, Joel Spencer, and J. Michael Steele, editors, *Discrete Probability and Algorithms*, pages 111–131. Springer New York, 1995.