On the Stretch Factor of Polygonal Chains

Ke Chen
Department of Computer Science, University of Wisconsin–Milwaukee, USA
kechen@uwm.edu

Adrian Dumitrescu
Department of Computer Science, University of Wisconsin–Milwaukee, USA
dumitres@uwm.edu

Wolfgang Mulzer
Institut für Informatik, Freie Universität Berlin, Germany
mulzer@inf.fu-berlin.de

Csaba D. Tóth
Department of Mathematics, California State University Northridge, Los Angeles, CA; and
Department of Computer Science, Tufts University, Medford, MA, USA
csaba.toth@csun.edu

Abstract

Let \( P = (p_1, p_2, \ldots, p_n) \) be a polygonal chain. The \textit{stretch factor} of \( P \) is the ratio between the total length of \( P \) and the distance of its endpoints, \( \sum_{i=1}^{n-1} |p_ip_{i+1}|/|p_1p_n| \). For a parameter \( c \geq 1 \), we call \( P \) a \( c \)-chain if \( |p_ip_j| + |p_jp_k| \leq c|p_ip_k| \), for every triple \((i, j, k)\), \(1 \leq i < j < k \leq n\). The stretch factor is a global property: it measures how close \( P \) is to a straight line, and it involves all the vertices of \( P \); being a \( c \)-chain, on the other hand, is a fingerprint-property: it only depends on subsets of \( O(1) \) vertices of the chain.

We investigate how the \( c \)-chain property influences the stretch factor in the plane: (i) we show that for every \( \varepsilon > 0 \), there is a noncrossing \( c \)-chain that has stretch factor \( \Omega\left(n^{1/2-\varepsilon}\right) \), for sufficiently large constant \( c = c(\varepsilon) \); (ii) on the other hand, the stretch factor of a \( c \)-chain \( P \) is \( O\left(n^{1/2}\right) \), for every constant \( c \geq 1 \), regardless of whether \( P \) is crossing or noncrossing; and (iii) we give a randomized algorithm that can determine, for a polygonal chain \( P \) in \( \mathbb{R}^2 \) with \( n \) vertices, the minimum \( c \geq 1 \) for which \( P \) is a \( c \)-chain in \( O\left(n^{2.5}\log\log n\right) \) expected time and \( O(n \log n) \) space.

2012 ACM Subject Classification
Mathematics of computing → Paths and connectivity problems;
Theory of computation → Computational geometry

Keywords and phrases
polygonal chain, vertex dilation, Koch curve, recursive construction

Abstract

Let \( P = (p_1, p_2, \ldots, p_n) \) be a polygonal chain. The stretch factor of \( P \) is the ratio between the total length of \( P \) and the distance of its endpoints, \( \sum_{i=1}^{n-1} |p_ip_{i+1}|/|p_1p_n| \). For a parameter \( c \geq 1 \), we call \( P \) a \( c \)-chain if \( |p_ip_j| + |p_jp_k| \leq c|p_ip_k| \), for every triple \((i, j, k)\), \(1 \leq i < j < k \leq n\). The stretch factor is a global property: it measures how close \( P \) is to a straight line, and it involves all the vertices of \( P \); being a \( c \)-chain, on the other hand, is a fingerprint-property: it only depends on subsets of \( O(1) \) vertices of the chain.

We investigate how the \( c \)-chain property influences the stretch factor in the plane: (i) we show that for every \( \varepsilon > 0 \), there is a noncrossing \( c \)-chain that has stretch factor \( \Omega\left(n^{1/2-\varepsilon}\right) \), for sufficiently large constant \( c = c(\varepsilon) \); (ii) on the other hand, the stretch factor of a \( c \)-chain \( P \) is \( O\left(n^{1/2}\right) \), for every constant \( c \geq 1 \), regardless of whether \( P \) is crossing or noncrossing; and (iii) we give a randomized algorithm that can determine, for a polygonal chain \( P \) in \( \mathbb{R}^2 \) with \( n \) vertices, the minimum \( c \geq 1 \) for which \( P \) is a \( c \)-chain in \( O\left(n^{2.5}\log\log n\right) \) expected time and \( O(n \log n) \) space.

Introduction

Given a set \( S \) of \( n \) point sites in the plane, what is the best way to connect \( S \) into a geometric network (graph)? This question has motivated researchers for a long time, going back as far as the 1940s, and beyond [19, 35]. Numerous possible criteria for a good geometric network
On the Stretch Factor of Polygonal Chains

have been proposed, perhaps the most basic being the length. In 1955, Few [20] showed that
for any set of \( n \) points in a unit square, there is a traveling salesman tour of length at most
\( \sqrt{2n} + 7/4 \). This was improved to at most \( 0.984\sqrt{2n} + 11 \) by Karloff [23]. Similar bounds
also hold for the shortest spanning tree and the shortest rectilinear spanning tree [13,16,21].
Besides length, two further key factors in the quality of a geometric network are the vertex
dilation and the geometric dilation [31], both of which measure how closely shortest paths in
a network approximate the Euclidean distances between their endpoints.

The dilation (also called stretch factor [29] or detour [1]) between two points \( p \) and \( q \) in a
geometric graph \( G \) is defined as the ratio between the length of a shortest path from \( p \) to \( q \)
and the Euclidean distance \( |pq| \). The dilation of the graph \( G \) is the maximum dilation over
all pairs of vertices in \( G \). A graph in which the dilation is bounded above by \( t \geq 1 \) is also
called a \( t \)-spanner (or simply a spanner if \( t \) is a constant). A complete graph in Euclidean
space is clearly a 1-spanner. Therefore, researchers focused on the dilation of graphs with
certain additional constraints, for example, noncrossing (i.e., plane) graphs. In 1989, Das
and Joseph [15] identified a large class of plane spanners (characterized by two simple local
properties). Bose et al. [6] gave an algorithm that constructs for any set of planar sites
a plane 11-spanner with bounded degree. On the other hand, Eppstein [18] analyzed a
fractal construction showing that \( \beta \)-skeletons, a natural class of geometric networks, can
have arbitrarily large dilation.

The study of dilation also raises algorithmic questions. Agarwal et al. [1] described
randomized algorithms for computing the dilation of a given path (on \( n \) vertices) in \( \mathbb{R}^2 \) in
\( O(n \log n) \) expected time. They also presented randomized algorithms for computing the
dilation of a given tree, or cycle, in \( \mathbb{R}^2 \) in \( O(n \log^2 n) \) expected time. Previously, Narasimhan
and Smid [30] showed that an \((1 + \varepsilon)\)-approximation of the stretch factor of any path, cycle,
or tree can be computed in \( O(n \log n) \) time. Klein et al. [24] gave randomized algorithms for
a path, tree, or cycle in \( \mathbb{R}^2 \) to count the number of vertex pairs whose dilation is below a
given threshold in \( O(n^{3/2 + \varepsilon}) \) expected time. Cheong et al. [12] showed that it is NP-hard to
determine the existence of a spanning tree on a planar point set whose dilation is at most a
given value. More results on plane spanners can be found in the monograph dedicated to
this subject [31] or in several surveys [8,17,29].

We investigate a basic question about the dilation of polygonal chains. More precisely,
we ask how the dilation between the endpoints of a polygonal chain (which we will call
the stretch factor, to distinguish it from the more general notion of dilation) is influenced
by fingerprint properties of the chain, i.e., by properties that are defined on \( O(1) \)-size
subsets of the vertex set. Such fingerprint properties play an important role in geometry,
where classic examples include the Carathéodory property\(^1\) [26, Theorem 1.2.3] or the Helly
property\(^2\) [26, Theorem 1.3.2]. In general, determining the effect of a fingerprint property
may prove elusive: given \( n \) points in the plane, consider the simple property that every 3
points determine 3 distinct distances. It is unknown [9, p. 203] whether this property implies
that the total number of distinct distances grows superlinearly in \( n \).

Furthermore, fingerprint properties appear in the general study of local versus global
properties of metric spaces that is highly relevant to combinatorial approximation algorithms
that are based on mathematical programming relaxations [5]. In the study of dilation,

---

\(^1\) Given a finite set \( S \) of points in \( d \) dimensions, if every \( d + 2 \) points in \( S \) are in convex position, then \( S \)
is in convex position.

\(^2\) Given a finite collection of convex sets in \( d \) dimensions, if every \( d + 1 \) sets have nonempty intersection,
then all sets have nonempty intersection.
interesting fingerprint properties have also been found. For example, a (continuous) curve $C$ is said to have the increasing chord property \cite{14,25} if for any points $a$, $b$, $c$, $d$ that appear on $C$ in this order, we have $|ad| \geq |bc|$. The increasing chord property implies that $C$ has (geometric) dilation at most $2\pi/3$ \cite{33}. A weaker property is the self-approaching property: a (continuous) curve $C$ is self-approaching if for any points $a$, $b$, $c$ that appear on $C$ in this order, we have $|ac| \geq |bc|$. Self-approaching curves have dilation at most 5.332 \cite{22} (see also \cite{3}), and they have found interesting applications in the field of graph drawing \cite{4,7,32}.

We introduce a new natural fingerprint property and see that it can constrain the stretch factor of a polygonal chain, but only in a weaker sense than one may expect; we also provide algorithmic results on this property. Before providing details, we give a few basic definitions.

**Definitions.** A polygonal chain $P$ in the Euclidean plane is specified by a sequence of $n$ points $(p_1, p_2, \ldots, p_n)$, called its vertices. The chain $P$ consists of $n-1$ line segments between consecutive vertices. We say $P$ is simple if only consecutive line segments intersect and they only intersect at their endpoints. Given a polygonal chain $P$ in the plane with $n$ vertices and a parameter $c \geq 1$, we call $P$ a $c$-chain if for all $1 \leq i < j < k \leq n$, we have

$$|p_ip_j| + |p_jp_k| \leq c|p_ip_k|. \quad (1)$$

Observe that the $c$-chain condition is a fingerprint condition that is not really a local dilation condition—it is more a combination between the local chain substructure and the distribution of the points in the subchains.

The stretch factor $\delta_P$ of $P$ is defined as the dilation between the two end points $p_1$ and $p_n$ of the chain:

$$\delta_P = \frac{\sum_{i=1}^{n-1} |p_ip_{i+1}|}{|p_1p_n|}.$$  

Note that this definition is different from the more general notion of dilation (also called stretch factor \cite{29}) of a graph which is the maximum dilation over all pairs of vertices. Since there is no ambiguity in this paper, we will just call $\delta_P$ the stretch factor of $P$.

For example, the polygonal chain $P = ((0, 0), (1, 0), \ldots, (n, 0))$ is a 1-chain with stretch factor 1; and $Q = ((0, 0), (0, 1), (1, 1), (1, 0))$ is a $(\sqrt{2} + 1)$-chain with stretch factor 3.

Without affecting the results, the floor and ceiling functions are omitted in our calculations.

For a positive integer $t$, let $[t] = \{1, 2, \ldots, t\}$. For a point set $S$, let $\text{conv}(S)$ denote the convex hull of $S$. All logarithms are in base 2, unless stated otherwise.

**Our results.** We deduce three upper bounds on the stretch factor of a $c$-chain $P$ with $n$ vertices (Section 2). In particular, we have (i) $\delta_P \leq c(n-1)\log c$, (ii) $\delta_P \leq c(n-2) + 1$, and (iii) $\delta_P = O\left(c^2 \sqrt{n - 1}\right)$.

From the other direction, we obtain the following lower bound (Section 3): For every $c \geq 4$, there is a family $P_c = \{P^k\}_{k \in \mathbb{N}}$ of simple $c$-chains, so that $P^k$ has $n = 4^k + 1$ vertices and stretch factor $(n-1)^{1 + \log(4^k+2)/\log 4}$, where the exponent converges to 1/2 as $c$ tends to infinity. The lower bound construction does not extend to the case of $1 < c < 4$, which remains open.

Finally, we present two algorithmic results (Section 4): (i) A randomized algorithm that decides, given a polygonal chain $P$ in $\mathbb{R}^2$ with $n$ vertices and a threshold $c > 1$, whether $P$ is a $c$-chain in $O\left(n^{2.5} \text{ polylog } n\right)$ expected time and $O(n \log n)$ space. (ii) As a corollary, there is a randomized algorithm that finds, for a polygonal chain $P$ with $n$ vertices, the minimum $c \geq 1$ for which $P$ is a $c$-chain in $O\left(n^{2.5} \text{ polylog } n\right)$ expected time and $O(n \log n)$ space.
At first glance, one might expect the stretch factor of a $c$-chain, for $c \geq 1$, to be bounded by some function of $c$. For example, the stretch factor of a 1-chain is necessarily 1. We derive three upper bounds on the stretch factor of a $c$-chain with $n$ vertices in terms of $c$ and $n$ (cf. Theorems 1–3); see Fig. 1 for a visual comparison between the bounds. For large $n$, the bound in Theorem 1 is the best for $1 \leq c \leq 2^{1/2}$, while the bound in Theorem 3 is the best for $c > 2^{1/2}$. In particular, the bound in Theorem 1 is tight for $c = 1$. The bound in Theorem 2 is the best for $c \geq 2$ and $n \leq 111c^2$.

Our first upper bound is obtained by a recursive application of the $c$-chain property. It holds for any positive distance function that may not even satisfy the triangle inequality.

\textbf{Theorem 1.} For a $c$-chain $P$ with $n$ vertices, we have $\delta_P \leq c(n - 1)\log c$.

\textbf{Proof.} We prove, by induction on $n$, that

$$\delta_P \leq c^{[\log(n-1)]},$$

for every $c$-chain $P$ with $n \geq 2$ vertices. In the base case, $n = 2$, we have $\delta_P = 1$ and $c^{[\log(2-1)]} = 1$. Now let $n \geq 3$, and assume that (2) holds for every $c$-chain with fewer than $n$ vertices. Let $P = (p_1, \ldots, p_n)$ be a $c$-chain with $n$ vertices. Then, applying (2) to the first and second half of $P$, followed by the $c$-chain property for the first, middle, and last vertex of $P$, we get

$$\sum_{i=1}^{n-1} |p_i p_{i+1}| \leq \sum_{i=1}^{[n/2]-1} |p_i p_{i+1}| + \sum_{i=[n/2]}^{n-1} |p_i p_{i+1}|$$

$$\leq c^{[\log([n/2]-1)]} (|p_1 p_{[n/2]}| + |p_{[n/2]} p_n|)$$

$$\leq c^{[\log([n/2]-1)]} \cdot c|p_1 p_n|$$

$$\leq c^{[\log(n-1)]}|p_1 p_n|,$$

so (2) holds also for $P$. Consequently,

$$\delta_P \leq c^{[\log(n-1)]} \leq c^{\log(n-1)+1} = c \cdot c^{\log(n-1)} = c(n - 1)\log c,$$

as required.
Our second bound interprets the $c$-chain property geometrically and makes use of the fact that $P$ resides in the Euclidean plane.

**Theorem 2.** For a $c$-chain $P$ with $n$ vertices, we have $\delta_P \leq c(n - 2) + 1$.

![Figure 2 The entire chain $P$ lies in an ellipse with foci $p_1$ and $p_n$.](image)

**Proof.** Without loss of generality, assume that $|p_1p_n| = 1$. Since $P$ is a $c$-chain, for every $1 < j < n$, we have $|p_jp_{j+1}| \leq |p_jp_n| = c$. If we fix the points $p_1$ and $p_n$, then every $p_j$ lies in an ellipse $E$ with foci $p_1$ and $p_n$, for $1 < j < n$, see Figure 2. The diameter of $E$ is its major axis, whose length is $c$. Since $E$ contains all vertices of the chain $P$, we have $|p_1p_2|, |p_{n-1}p_n| \leq \frac{c}{2} \leq c$ and $|p_jp_{j+1}| \leq c$ for all $1 < j < n - 1$. Therefore the stretch factor of $P$ is bounded above by

$$
\delta_P = \frac{\sum_{j=1}^{n-1} |p_jp_{j+1}|}{|p_1p_n|} = |p_1p_2| + |p_{n-1}p_n| + \sum_{j=2}^{n-2} |p_jp_{j+1}|
\leq \frac{c + 1}{2} + \frac{c + 1}{2} + c(n - 3) = c(n - 2) + 1,
$$

as required.

Our third upper bound uses a volume argument to bound the number of long edges in $P$.

**Theorem 3.** Let $P = (p_1, \ldots, p_n)$ be a $c$-chain, for some constant $c \geq 1$, and let $L = \sum_{i=1}^{n-1} |p_ip_{i+1}|$ be its length. Then $L = O\left(c^2\sqrt{n-1}\right) |p_1p_n|$, hence $\delta_P = O\left(c^2\sqrt{n-1}\right)$.

**Proof.** We may assume that $p_1p_n$ is a horizontal segment of unit length. By the argument in the proof of Theorem 2, all points $p_i$ ($i = 1, \ldots, n$) are contained in an ellipse $E$ with foci $p_1$ and $p_n$, where the major axis of $E$ has length $c$. Let $U$ be the minimal axis-aligned square containing $E$; its side is of length $c$.

We set $x = 8c^2/\sqrt{n-1}$; and let $L_0$ and $L_1$ be the sum of lengths of all edges in $P$ of length at most $x$ and more than $x$, respectively. By definition, we have $L = L_0 + L_1$ and

$$
L_0 \leq (n - 1)x = (n - 1) \cdot 8c^2/\sqrt{n-1} = 8c^2\sqrt{n-1}.
$$

We shall prove that $L_1 \leq 8c^2\sqrt{n-1}$, implying $L \leq 2x(n-1) = O\left(c^2\sqrt{n-1}\right)$. For this, we further classify the edges in $L_1$ according to their lengths: For $\ell = 0, 1, \ldots, \infty$, let

$$
P_\ell = \{p_i : 2^\ell x < |p_ip_{i+1}| \leq 2^{\ell+1}x\}.
$$

Since all points lie in an ellipse of diameter $c$, we have $|p_ip_{i+1}| \leq c$, for all $i = 0, \ldots, n - 1$. Consequently, $P_\ell = \emptyset$ when $c \leq 2^\ell x$, or equivalently $\log(c/x) \leq \ell$. 


We use a volume argument to derive an upper bound on the cardinality of $P_c$ for $\ell = 0, 1, \ldots, \lfloor \log(c/x) \rfloor$. Assume that $p_i, p_k \in P_c$, and w.l.o.g., $i < k$. If $k = i + 1$, then by (4), $2^\ell x < |p_i p_k|$. Otherwise,

$$2^\ell x < |p_i p_{i+1}| < |p_i p_{i+1} + |p_{i+1} p_k| \leq c|p_i p_k|,$$

or $2^\ell x/c < |p_i p_k|$.

Consequently, the disks of radius

$$R = \frac{2^\ell x}{2c} = \frac{4 \cdot 2^\ell c}{\sqrt{n-1}} \tag{5}$$

centered at the points in $P_c$ are interior-disjoint. The area of each disk is $\pi R^2$. Since $P_c \subset U$, these disks are contained in the $R$-neighborhood $U_R$ of the square $U$, i.e., the Minkowski sum $R + U$. For $\ell \leq \log(c/x)$, we have $2^\ell x \leq c$, hence $R = \frac{2^\ell x}{2c} = \frac{c}{2} = \frac{1}{2} \leq \frac{c}{2}$. Then we can bound the area of $U_R$ from above as follows:

$$\text{area}(U_R) < (c + 2R)^2 \leq (2c)^2 = 4c^2. \tag{6}$$

Since $U_R$ contains $|P_c|$ interior-disjoint disks of radius $R$, we obtain

$$|P_c| \leq \frac{\text{area}(U_R)}{\pi R^2} < \frac{4c^2}{\pi R^2} = \frac{16c^4}{\pi 2^{2\ell} x^2}. \tag{7}$$

For every segment $p_{i-1} p_i$ with length more than $x$, we have that $p_i \in P_c$, for some $\ell \in \{0, 1, \ldots, \lfloor \log(c/x) \rfloor\}$. The total length of these segments is

$$L_1 \leq \sum_{\ell=0}^{\lfloor \log(c/x) \rfloor} |P_c| \cdot 2^{\ell+1} x < \sum_{\ell=0}^{\lfloor \log(c/x) \rfloor} \frac{16c^4}{\pi 2^{2\ell} x^2} \cdot 2^{\ell+1} x = \sum_{\ell=0}^{\lfloor \log(c/x) \rfloor} \frac{32c^4}{\pi 2^\ell x} \leq \frac{32c^4}{\pi x} \sum_{\ell=0}^{\infty} \frac{1}{2^\ell} = \frac{64c^4}{\pi x} = \frac{8c^2}{\pi} \cdot \sqrt{n-1},$$

as required. Together with (3), this yields $L \leq 8 \left(1 + \frac{c^2}{\pi}\right) \cdot \sqrt{n-1}$. \hfill \blacktriangleup

### 3 Lower Bounds

We now present our lower bound construction, showing that the dependence on $n$ for the stretch factor of a $c$-chain cannot be avoided.

**Theorem 4.** For every constant $c \geq 4$, there is a set $P_c = \{P^k\}_{k \in \mathbb{N}}$ of simple $c$-chains, so that $P^k$ has $n = 4^k + 1$ vertices and stretch factor $(n-1)^{1/2 - \log c/2}$. By Theorem 3, the stretch factor of a $c$-chain in the plane is $O((n-1)^{1/2})$ for every constant $c \geq 1$. Since

$$\lim_{c \to \infty} \frac{1 + \log(c-2) - \log c}{2} = \frac{1}{2},$$

our lower bound construction shows that the limit of the exponent cannot be improved.

Indeed, for every $\varepsilon > 0$, we can set $c = \frac{2^{2\varepsilon+1}}{2^{2\varepsilon-1} - 1}$, and then the chains above have stretch factor $(n-1)^{1/2 - \log c/2} = (n-1)^{1/2 - \varepsilon} = \Omega(n^{1/2-\varepsilon})$.

We first construct a family $P_c = \{P^k\}_{k \in \mathbb{N}}$ of polygonal chains. Then we show, in Lemmata 5 and 6, that every chain in $P_c$ is simple and indeed a $c$-chain. The theorem follows since the claimed stretch factor is a consequence of the construction.
Construction of $\mathcal{P}_c$. The construction here is a generalization of the iterative construction of the Koch curve; when $c = 6$, the result is the original Cesàro fractal (which is a variant of the Koch curve) [10]. We start with a unit line segment $P^0$, and for $k = 0, 1, \ldots$, we construct $P^{k+1}$ by replacing each segment in $P^k$ by four segments such that the middle three points achieve a stretch factor of $c_a = \frac{c-2}{2}$ (this choice will be justified in the proof of Lemma 6). Note that $c_a \geq 1$, since $c \geq 4$.

We continue with the details. Let $P^0$ be the unit line segment from $(0, 0)$ to $(1, 0)$; see Figure 3 (left). Given the polygonal chain $P^k$ ($k = 0, 1, \ldots$), we construct $P^{k+1}$ by replacing each segment of $P^k$ by four segments as follows. Consider a segment of $P^k$, and denote its length by $\ell$. Subdivide this segment into three segments of lengths $(\frac{1}{2} - \frac{a}{c_a})\ell$, $\frac{2a}{c_a}\ell$, and $(\frac{1}{2} - \frac{a}{c_a})\ell$, respectively, where $0 < a < \frac{c}{2}$ is a parameter to be determined later. Replace the middle segment with the top part of an isosceles triangle of side length $a\ell$. The chains $P^0$, $P^1$, $P^2$, and $P^4$ are depicted in Figures 3 and 4.

Note that each segment of length $\ell$ in $P^k$ is replaced by four segments of total length $(1 + \frac{2a(c_a-1)}{c_a})\ell$. After $k$ iterations, the chain $P^k$ consists of $4^k$ line segments of total length $(1 + \frac{2a(c_a-1)}{c_a})^k$.

By construction, the chain $P^k$ (for $k \geq 1$) consists of four scaled copies of $P^{k-1}$. For $i = 1, 2, 3, 4$, let the $i$th subchain of $P^k$ be the subchain of $P^k$ consisting of $4^{k-1}$ segments starting from the ($(i-1)4^{k-1} + 1$)th segment. By construction, the $i$th subchain of $P^k$ is similar to the chain $P^{k-1}$, for $i = 1, 2, 3, 4$. The following functions allow us to refer to these subchains formally. For $i = 1, 2, 3, 4$, define a function $f^k_i : P^k \rightarrow P^k$ as the identity on the $i$th subchain of $P^k$ that sends the remaining part(s) of $P^k$ to the closest endpoint(s) along this subchain. So $f^k_i(P^k)$ is similar to $P^{k-1}$. Let $g_i : \mathcal{P}_c \setminus \{P^0\} \rightarrow \mathcal{P}_c$ be a piecewise defined function such that $g_i(C) = \sigma^{-1} \circ f^k_i \circ \sigma(C)$ if $C$ is similar to $P^k$, where $\sigma : C \rightarrow P^k$ is a similarity transformation. Applying the function $g_i$ on a chain $P^k$ can be thought of as “cutting out” its $i$th subchain.

Figure 3 The chains $P^0$ (left) and $P^1$ (right).

Figure 4 The chains $P^2$ (left) and $P^4$ (right).

---

3 Two geometric shapes are similar if one can be obtained from the other by translation, rotation, and scaling; and are congruent if one can be obtained from the other by translation and rotation.
Clearly, the stretch factor of the chain monotonically increases with the parameter \( a \).

However, if \( a \) is too large, the chain is no longer simple. The following lemma gives a sufficient condition for the constructed chains to avoid self-crossings.

\[ \textbf{Lemma 5. For every constant } c \geq 4, \text{ if } a \leq \frac{c - 2}{2c}, \text{ then every chain in } \mathcal{P}_c \text{ is simple.} \]

\[ \textbf{Proof.} \text{ Let } T = \text{conv}(P^1). \text{ Observe that } T \text{ is an isosceles triangle; see Figure 5 (left). We first show the following:} \]

\[ \triangledown \text{ Claim. If } a \leq \frac{c - 2}{2c}, \text{ then conv}(P^k) = T \text{ for all } k \geq 1. \]

\[ \text{Proof.} \text{ We prove the claim by induction on } k. \text{ It holds for } k = 1 \text{ by definition. For the induction step, assume that } k \geq 2 \text{ and that the claim holds for } k - 1. \text{ Consider the chain } P^k. \text{ Since it contains all the vertices of } P^1, T \subset \text{conv}(P^k). \text{ So we only need to show that } \text{conv}(P^k) \subset T. \]

\[ \text{Figure 5} \text{ Left: Convex hull } T \text{ of } P^1 \text{ in light gray; Right: Convex hulls of } g_i(P^k), i = 1, 2, 3, 4, \text{ in dark gray, are contained in } T. \]

By construction, \( P^k \subset \bigcup_{i=1}^4 \text{conv}(g_i(P^k)); \text{ see Figure 5 (right).} \) By the inductive hypothesis, \( \text{conv}(g_i(P^k)) \) is an isosceles triangle similar to \( T \), for \( i = 1, 2, 3, 4 \). Since the bases of \( \text{conv}(g_1(P^k)) \) and \( \text{conv}(g_4(P^k)) \) are collinear with the base of \( T \) by construction, due to similarity, they are contained in \( T \). The base of \( \text{conv}(g_2(P^k)) \) is contained in \( T \). In order to show \( \text{conv}(g_1(P^k)) \subset T \), by convexity, it suffices to ensure that its apex \( p \) is also in \( T \). Note that the coordinates of the top point is \( t = \left( \frac{1}{2}, a \sqrt{c_x^2 - 1}/c_x \right) \), so the supporting line \( \ell \) of the left side of \( T \) is

\[ y = \frac{2a \sqrt{c_x^2 - 1}}{c_x} x, \text{ and} \]

\[ p = \left( \frac{1}{2} - \frac{a}{2c_x} - \frac{a^2(c_x^2 - 1)}{c_x^2}, \left( \frac{a}{2c_x^2} + \frac{a^2}{c_x^2} \right) \sqrt{c_x^2 - 1} \right). \]

By the condition of \( a \leq \frac{c - 2}{2c} \), the apex \( p \) lies on or below \( \ell \). Under the same condition, we have \( \text{conv}(g_1(P^k)) \subset T \) by symmetry. Then \( P^k \subset \bigcup_{i=1}^4 \text{conv}(g_i(P^k)) \subset T. \) Since \( T \) is convex, \( \text{conv}(P^k) \subset T \). So \( \text{conv}(P^k) = T \), as claimed. \[ \triangledown \]

We can now finish the proof of Lemma 5 by induction. Clearly, \( P^0 \) and \( P^1 \) are simple. Assume that \( k \geq 2, \text{ and } P^{k-1} \) is simple. Consider the chain \( P^k. \) For \( i = 1, 2, 3, 4, \) \( g_i(P^k) \) is similar to \( P^{k-1} \), hence simple by the inductive hypothesis. Since \( P^k = \bigcup_{i=1}^4 g_i(P^k) \), it is sufficient to show that for all \( i, j \in \{1, 2, 3, 4\}, \text{ where } i \neq j, \text{ a segment in } g_j(P^k) \text{ does not intersect any segments in } g_j(P^k), \text{ unless they are consecutive in } P^k \text{ and they intersect at a common endpoint. This follows from the above claim together with the observation that for } i \neq j, \text{ the intersection } g_i(P^k) \cap g_j(P^k) \text{ is either empty or contains a single vertex which is the common endpoint of two consecutive segments in } P^k. \]
In the remainder of this section, we assume that
\[ a = \frac{c - 2}{2c} = \frac{c_*}{2(c_* + 1)}. \] (8)
Under this assumption, all segments in \( P^1 \) have the same length \( a \). Therefore, by construction, all segments in \( P^k \) have the same length
\[ a^k = \left( \frac{c_*}{2(c_* + 1)} \right)^k. \]
There are \( 4k \) segments in \( P^k \), with \( 4k + 1 \) vertices, and its stretch factor is
\[ \delta_{P^k} = 4^k \left( \frac{c_*}{2(c_* + 1)} \right)^k = \left( \frac{2c_*}{c_* + 1} \right)^k. \]
Consequently, \( k = \log_4(n - 1) = \frac{\log(n - 1)}{2} \), and
\[ \delta_{P^k} = \left( \frac{2c_*}{c_* + 1} \right)^{\frac{\log(n - 1)}{2}} = \left( \frac{2c - 4}{c} \right)^{\frac{\log(n - 1)}{2}} = (n - 1)^{\frac{1 + \log(c - 2) - \log c}{2}}, \]
as claimed. To finish the proof of Theorem 4, it remains to show the constructed polygonal chains are indeed \( c \)-chains.

**Lemma 6.** For every constant \( c \geq 4 \), \( \mathcal{P}_c \) is a family of \( c \)-chains.

We first prove a couple of facts that will be useful in the proof of Lemma 6. We defer an intuitive explanation until after the formal statement of the lemma.

**Lemma 7.** Let \( k \geq 1 \) and let \( P^k = (p_1, p_2, \ldots, p_n) \), where \( n = 4^k + 1 \). Then the following hold:

(i) There exists a sequence \((q_1, q_2, \ldots, q_m)\) of \( m = 2 \cdot 4^{k-1} \) points in \( \mathbb{R}^2 \) such that the chain \( R^k = (p_1, q_1, p_2, q_2, \ldots, p_m, q_m, p_{m+1}) \) is similar to \( P^k \).

(ii) For \( k \geq 2 \), define \( g_5 : \mathcal{P}_c \setminus \{P^0, P^1\} \to \mathcal{P}_c \) by
\[ g_5(P^k) = (g_3 \circ g_2(P^k)) \cup (g_4 \circ g_2(P^k)) \cup (g_1 \circ g_3(P^k)) \cup (g_2 \circ g_3(P^k)). \]
Then \( g_5(P^k) \) is similar to \( P^{k-1} \).

Part (i) of Lemma 7 says that given \( P^k \), we can construct a chain \( R^k \) similar to \( P^k \) by inserting one point between every two consecutive points of the left half of \( P^k \), see Figure 6(left). Part (ii) says that the “top” subchain of \( P^k \) that consists of the right half of \( g_2(P^k) \) and the left half of \( g_3(P^k) \), see Figure 6(right), is similar to \( P^{k-1} \).

**Figure 6** Left: Chain \( P^k \) with the scaled copy of itself \( R^k \) (in red); Right: Chain \( P^k \) with its subchain \( g_5(P^k) \) marked by its convex hull.
Proof of Lemma 7. For (i), we review the construction of $P^k$, and show that $R^k$ and $P^k$
can be constructed in a coupled manner. In Figure 7(left), consider $P^1 = (p_1, p_2, p_3, p_4, p_5)$.
Recall that all segments in $P^1$ are of the same length $a = \frac{\pi}{2\sqrt{17}}$. The isosceles triangles
$\Delta p_1 p_2 p_3$ and $\Delta p_1 p_3 p_5$ are similar. Let $\sigma : \Delta p_1 p_3 p_5 \rightarrow \Delta p_1 p_2 p_3$ be the similarity transformation. Let $q_1 = \sigma(p_2)$ and $q_2 = \sigma(p_1)$. By construction, the chain $R^1 = (p_1, q_1, p_2, q_2, p_3)$
is similar to $P^1$. In particular, all of its segments have the same length. So the isosceles
triangle $\Delta p_1 q_1 p_2$ is similar to $\Delta p_1 p_3 p_5$. Moreover, its base is the segment $p_1 p_2$, so $\Delta p_1 q_1 p_2$
is precisely $\text{conv}(q_1(P^2))$, see Figure 7(right).

![Figure 7](Left: the chains $P^1$ and $R^1$ (red); Right: the chains $P^2$ and $R^1$ (red).)

Write $P^2 = (v_1, v_2, \ldots, v_{17})$, then $v_3 = q_1$ by the above argument and $v_7 = q_2$ by
symmetry. Now $\Delta v_1 v_2 v_3$, $\Delta v_3 v_4 v_5$, $\Delta v_5 v_6 v_7$, and $\Delta v_7 v_8 v_9$ are four congruent isosceles
triangles, all of which are similar to $\Delta v_1 v_9 v_{17}$, since the angles are the same. Repeat the
above procedure on each of them to obtain $R^2 = (v_1, u_1, v_2, u_2, \ldots, v_8, u_8, v_9)$, which is similar
to $P^2$. Continue this construction inductively to get the desired chain $R^k$ for any $k \geq 1$.

For (ii), see Figure 7(right). By definition, $g_k(P^2)$ is the subchain $(v_7, v_8, v_9, v_{10}, v_{11})$.
Observe that the segments $v_7 v_8$ and $v_{10}v_{11}$ are collinear by symmetry. Moreover, they are
parallel to $v_{11}v_{17}$ since $\angle v_7 v_8 v_9 = \angle v_{10} v_{11} v_{17}$. So $g_k(P^2)$ is similar to $P^1$; see Figure 7(left).
Then for $k \geq 2$, $g_k(P^k)$ is the subchain of $P^k$ starting at vertex $v_7$, ending at vertex $v_{11}$. By
the construction of $P^k$, $g_k(P^k)$ is similar to $P^{k-1}$.

Due to space constraints, the proof of Lemma 6 is deferred to the full version.

4 Algorithm for Recognizing $c$-Chains

In this section, we design a randomized Las Vegas algorithm to recognize $c$-chains. More
precisely, given a polygonal chain $P = (p_1, \ldots, p_n)$, and a parameter $c \geq 1$, the algorithm
decides whether $P$ is a $c$-chain, in $O(n^{2.5} \log n)$ expected time. By definition, $P =
(p_1, \ldots, p_n)$ is a $c$-chain if $|p_i p_j| + |p_j p_k| \leq c |p_i p_k|$ for all $1 \leq i < j < k \leq n$; equivalently,
$p_j$ lies in the ellipse of major axis $c$ with foci $p_i$ and $p_k$. Consequently, it suffices to test,
for every pair $1 \leq i < k \leq n$, whether the ellipse of major axis $c|p_i p_k|$ with foci $p_i$ and $p_k$
contains $p_j$, for all $j$, $i < j < k$. For this, we can apply recent results from geometric range
searching.

Theorem 8. There is a randomized algorithm that can decide, for a polygonal chain
$P = (p_1, \ldots, p_n)$ in $\mathbb{R}^2$ and a threshold $c > 1$, whether $P$ is a $c$-chain in $O(n^{2.5} \log n)$
expected time and $O(n \log n)$ space.

Agarwal, Matoušek and Sharir [2, Theorem 1.4] constructed, for a set $S$ of $n$ points in
$\mathbb{R}^2$, a data structure that can answer ellipse range searching queries: it reports the number
of points in $S$ that are contained in a query ellipse. In particular, they showed that, for
every $\varepsilon > 0$, there is a constant $B$ and a data structure with $O(n)$ space, $O(n^{1+\varepsilon})$ expected
preprocessing time, and $O\left(n^{1/2} \log^B n\right)$ query time. The construction was later simplified
by Matoušek and Patáková [27]. Using this data structure, we can quickly decide whether a
given polygonal chain is a c-chain.

**Proof of Theorem 8.** Subdivide the polygonal chain \( P = (p_1, \ldots, p_n) \) into two subchains of
equal or almost equal sizes, \( P_1 = (p_1, \ldots, p_{n/2}) \) and \( P_2 = (p_{n/2}, \ldots, p_n) \); and recursively
subdivide \( P_1 \) and \( P_2 \) until reaching 1-vertex chains. Denote by \( T \) the recursion tree. Then,
\( T \) is a binary tree of depth \( \lceil \log n \rceil \). There are at most \( 2^i \) nodes at level \( i \); the nodes at level \( i \)
correspond to edge-disjoint subchains of \( P \), each of which has at most \( n/2^i \) edges. Let \( W_i \) be
the set of subchains on level \( i \) of \( T \); and let \( W = \bigcup_{i \geq 0} W_i \). We have \( |W| \leq 2n \).

For each polygonal chain \( Q \in W \), construct an ellipse range searching data structure
\( DS(Q) \) described above [2] for the vertices of \( Q \), with a suitable parameter \( \varepsilon > 0 \). Their
overall expected preprocessing time is

\[
\sum_{i=0}^{[\log n]} 2^i \cdot O \left( \left( \frac{n}{2^i} \right)^{1+\varepsilon} \right) = O \left( n^{1+\varepsilon} \right),
\]

their space requirement is \( \sum_{i=0}^{[\log n]} 2^i \cdot O \left( n/2^i \right) = O(n \log n) \), and their query time at level \( i \)
is \( O \left( \left( n/2^i \right)^{1/2} \text{polylog} \left( n/2^i \right) \right) = O \left( n^{1/2} \text{polylog} n \right) \).

For each pair of indices \( 1 \leq i < k \leq n \), we do the following. Let \( E_{i,k} \) denote the ellipse of
major axis \( c|p_ip_k| \) with foci \( p_i \) and \( p_k \). The chain \( (p_{i+1}, \ldots, p_{k-1}) \) is subdivided into \( O(\log n) \)
maximal subchains in \( W \), using at most two subchains from each set \( W_i \), \( i = 0, \ldots, [\log n] \).

For each of these subchains \( Q \in W \), query the data structure \( DS(Q) \) with the ellipse \( E_{i,k} \). If
all queries are positive (i.e., the count returned is \( |Q| \) in all queries), then \( P \) is a c-chain;
otherwise there exists \( j, i < j < k \), such that \( p_j \notin E_{i,k} \), hence \( |p_i p_j| + |p_j p_k| > c|p_i p_k| \),
witnessing that \( P \) is not a c-chain.

The query time over all pairs \( 1 \leq i < k \leq n \) is bounded above by

\[
\left( \frac{n}{2} \right)^{2[\log n]} \sum_{i=0}^{[\log n]} O \left( \left( \frac{n}{2^i} \right)^{1/2} \text{polylog} \left( n/2^i \right) \right) = \left( \frac{n}{2} \right) \cdot O \left( n^{1/2} \text{polylog} n \right)
= O \left( n^{2.5} \text{polylog} n \right).
\]

This subsumes the expected time needed for constructing the structures \( DS(Q) \), for all
\( Q \in W \). So the overall running time of the algorithm is \( O \left( n^{2.5} \text{polylog} n \right) \), as claimed. ◀

In the decision algorithm above, only the construction of the data structures \( DS(Q) \),
\( Q \in W \), uses randomization, which is independent of the value of \( c \). The parameter \( c \) is used
for defining the ellipses \( E_{i,k} \), and the queries to the data structures; this part is deterministic.

Hence, we can find the optimal value of \( c \) by Meggido’s parametric search [28] in the second
part of the algorithm.

Meggido’s technique reduces an optimization problem to a corresponding decision problem
at a polylogarithmic factor increase in the running time. An optimization problem is amenable
to this technique if the following three conditions are met [34]: (1) the objective function
is monotone in the given parameter; (2) the decision problem can be solved by evaluating
bounded-degree polynomials, and (3) the decision problem admits an efficient parallel
algorithm (with polylogarithmic running time using polynomial number of processors). All
three conditions hold in our case: The area of each ellipse with foci in \( S \) monotonically
increases with \( c \); the data structure of [27] answers ellipse range counting queries by evaluating
polynomials of bounded degree; and the \( \binom{n}{2} \) queries can be performed in parallel. Alternatively,
Chan’s randomized optimization technique [11] is also applicable. Both techniques yield the following result.

**Corollary 9.** There is a randomized algorithm that can find, for a polygonal chain \( P = (p_1, \ldots, p_n) \) in \( \mathbb{R}^2 \), the minimum \( c \geq 1 \) for which \( P \) is a \( c \)-chain in \( O(n^{2.5} \log n) \) expected time and \( O(n \log n) \) space.

We remark that, for \( c = 1 \), the test takes \( O(n) \) time: it suffices to check whether points \( p_3, \ldots, p_n \) lie on the line spanned by \( p_1p_2 \), in that order.

### 5 Concluding Remarks

We end with some final observations and pointers for further research.

1. For \( k \geq 1 \), let \( P_k^c = g_2(P^k) \cup g_3(P^k) \), see Figure 8 (right). It is easy to see that \( P_k^c \) is a \( c \)-chain with \( n = 4^k/2 + 1 \) vertices and has stretch factor \( \sqrt{c(c - 2)/8(n - 1)} \). Since \( c(c - 2)/8 \geq 1 \) for \( c \geq 4 \), this improves the result of Theorem 4 by a constant factor. Since this construction does not improve the exponent, and the analysis would be longer (requiring a case analysis without new insights), we omit the details.

2. If \( c \) is used instead of \( c_* = (c - 2)/2 \) in the lower bound construction, then the condition \( c \geq 4 \) in Theorem 4 can be replaced by \( c \geq 1 \), and the bound can be improved from \((n - 1)^{1/2 - \epsilon} \) to \((n - 1)^{1/2} \). However, we were unable to prove that the resulting \( P^k \)'s, \( k \in \mathbb{N} \), are \( c \)-chains, although a computer program has verified that the first few generations of them are indeed \( c \)-chains.

3. The volume argument in Theorem 3 easily generalizes to higher dimensions. If \( P \) be a \( c \)-chain in \( \mathbb{R}^d \) for fixed \( c \geq 1 \) and \( d \geq 2 \), then \( \delta_P = O(\epsilon^2(n - 1)^{1-1/d}) \). It is interesting to find out whether extra dimension(s) allows one to achieve a larger stretch factor.

4. The upper bounds in Theorem 1–3 are valid regardless of whether the chain is crossing or not. On the other hand, the lower bound in Theorem 4 is given by noncrossing chains. A natural question is whether a sharper upper bound holds if the chains are required to be noncrossing. More specifically, can the exponent of \( n \) in the upper bound be reduced to \( 1/2 - \epsilon \), where \( \epsilon > 0 \) depends on \( c \)?

5. Our algorithm in Section 4 can recognize \( c \)-chains with \( n \) vertices in \( O(n^{2.5} \log n) \) expected time and \( O(n \log n) \) space, using ellipse range searching data structures. It is likely that the running time can be improved in the future, perhaps at the expense of increased space, when suitable time-space trade-offs for semi-algebraic range searching become available. The existence of such data structures is conjectured [2], but currently remains open.
References


On the Stretch Factor of Polygonal Chains


