A General Algorithm for the Maximum Span of Fixed-Angle Chains

David Goering†

Ronald Gentle‡

Abstract

Fixed-angle chains have been used to model protein backbones [4] and robotic arm motions [5]. Benbernou and O’Rourke proved several structural theorems for finding the maximum 3D span of fixed-angle chains: the largest distance achievable between the two endpoints [1][2]. Borcea and Streinu used different methods to develop a new algorithm which also computes the maximum span for any fixed-angle chain and the configuration in which this is achieved. Our algorithm is purely geometric in nature, meaning that it consists of only straight-edge and compass constructions together with some list-keeping. The algorithm has complexity $O(n^2)$ for any chain with equal angles, known also as $\alpha$-chains. We do not claim that it runs in polynomial time for all chains but discuss why it will do so for those likely to be used in any modeling application.

1 Introduction

Fixed-angle chains consist of serially connected line segments, each attached to its predecessor at an angle $0^\circ < \alpha_i < 180^\circ$ but capable of spinning at the joint while the angle between the two segments remains constant. Soss proved that finding the maximum span of flat configurations of the chain is NP-hard, but showed that the 3D maxspan is not always achieved by a flat configuration [3]. Benbernou and O’Rourke primarily focused on the maximum 3D span for restricted classes of chains. They conjectured that all $\alpha$-chains are solvable in quadratic time and our results verify this (it is possible that Borcea and Streinu also show this for chains with $\alpha < \pi/3$ but we are not aware of this result). Our algorithm directly depends on their $n$-Chain Partition Theorem which we state after introducing notation, most of which is consistent with [2].

Let a chain $C$ have vertices $(v_0, v_1, \ldots, v_n)$. The fixed joint angle is $\alpha_i = \angle v_{i-1}v_iv_{i+1}$. We denote link $i$ (the line segment $v_{i-1}v_i$) as $L_i$. A flat configuration for $C$ is one in which all vertices lie in the same plane. The zigzag or trans-configuration is the flat configuration in which the direction of the joint turns alternates. The chain $C$ is in maxspan configuration when it is positioned to maximize the distance $|v_0v_n|$. We refer to the position of $v_n$ in maxspan configuration as the maxpt.

**Theorem 1 (n-Chain Partition Theorem)** [2] The planar partition for an $n$-chain $C$ (described below) in maxspan configuration has the following two properties:

1. The vertices shared between adjacent planar sections all lie along the line $L$ through $v_0v_n$.

2. The last planar section cannot contain just one link $v_{n-1}v_n$.

This implies that in maxspan configuration the vertices $v_0$, $v_1$, $v_2$ and $v_n$ all lie in the same plane. Furthermore if the maxspan configuration is not flat, then the vertices can be partitioned as follows: “Group $v_0, \ldots, v_i$ into one section if they lie in plane $\Pi_1$, but $v_{i+1}$ does not lie in this plane. Then group $v_{i+1}, \ldots, v_j$ into a second section if they lie in plane $\Pi_2 \neq \Pi_1$, and $v_{j+1}$ does not lie in $\Pi_2$. And so on” [2]. The vertices $v_0$, $v_1$, $v_2$ and $v_n$ all lie on the same line, and therefore all lie in the plane $\Pi_1$. See Fig. 1.

![Figure 1: A 12-chain in maxspan configuration. Here vertices $v_0$, $v_2$, $v_4$, $v_6$, $v_8$, $v_{10}$ and $v_{12}$ are collinear.](image)

Our search for the maxpt begins by laying out $C$ in the zigzag configuration mentioned above. Note that some chains will be self-crossing when laid out this way and may possibly be self-crossing in the maxspan configuration as well. While these chains may not be of practical importance our method does not exclude this possibility.

The idea behind our algorithm is to search for the maxpt by systematically allowing the links to rotate out...
of \(\Pi_1\) beginning with \(L_n\), then \(L_{n-1}\), etc. At each step we observe which points in \(\Pi_1\) are reachable by \(v_n\) under all possible rotations of the free links. We then identify those points which could conceivably be the maxpt for \(C\) and exclude the rest from further consideration. At each step new points reachable by \(v_n\) will be found, and points found in previous iterations may be excluded.

2. The subchain \(C_{n-4} = (v_{n-4}, v_{n-3}, v_{n-2}, v_{n-1}, v_n)\)

We begin by allowing \(L_n\) to rotate out of \(\Pi_1\) about \(L_{n-1}\) while maintaining a constant angle \(\alpha_{n-1}\) at the point of attachment \(v_{n-1}\). As it does so, the locus of \(v_n\) is a circle orthogonal to \(\Pi_1\) centered at the projection of \(v_n\) onto \(L_{n-1}\) extended. This circle intersects \(\Pi_1\) in the original position of \(v_n\) and in it’s reflection about \(L_{n-1}\). We denote this reflection \(v_n(n-1)\). See the circle in Fig. 2. Note that for the subchain \(C_{n-3} = (v_{n-3}, v_{n-2}, v_{n-1}, v_n)\) the maxspan is \(|v_{n-3}v_n|\) and is achieved in the trans-configuration [1].

Now allow \(L_{n-1}\) to rotate similarly about \(L_{n-2}\) while also allowing \(L_n\) to rotate about \(L_{n-1}\). The points traced by \(v_n\) in this process comprise a partial sphere with center \(v_{n-2}\) and radius \(|v_{n-2}v_n|\). The intersection of this partial sphere with \(\Pi_1\) is two circular arcs, each also centered at \(v_{n-2}\) with radius \(|v_{n-2}v_n|\). See Fig. 2.

Observation 1 Assuming again that \(C\) is in trans-configuration there are two cases.

1. The points \(v_n\) and \(v_n(n-1)\) lie on the same side of \(L_{n-2}\) extended. Then this line does not pass through the circle created by the rotation of \(L_n\) and these are the arc endpoints on one side. Their reflections about \(L_{n-2}\) are the arc endpoints on the other side which we denote \(v_n(n-2)\) and \(v_n(n-1, n-2)\).

2. The points \(v_n\) and \(v_n(n-1)\) lie on opposite sides of \(L_{n-2}\). Then \(L_{n-2}\) extended goes through the circle created by the rotation of \(L_n\). In this case the arc endpoints on one side are \(v_n(n-1, n-2)\) and \(v_n\), with endpoints \(v_n(n-1)\) and \(v_n(n-2)\) on the other.

We illustrate these cases in Fig. 3.

![Figure 3: The two cases for arc endpoints following rotations of the last two links in the chain.](image)

Figure 3: The two cases for arc endpoints following rotations of the last two links in the chain.

We are now in a position to find the maxspan of \(C_{n-4}\). Note that we can do so without taking into account the points generated by the rotation of \(L_{n-2}\) at \(v_{n-3}\). This rotation would move any point on the partial sphere in a circle about \(L_{n-3}\), with each point on the circle remaining equidistant from any point on that line, specifically \(v_{n-4}\). So none of these new points would be farther from \(v_{n-4}\) than the points on the two arcs.

To find the point on the arcs farthest from \(v_{n-4}\) we use the following basic facts.

**Lemma 2** Let \(C\) be a circle with center \(B\), let \(A\) be an arc on \(C\), and let \(P\) be a point in the plane of \(C\) other than \(B\). The line \(PB\) intersects \(C\) in two points. Let \(Q\) be the farther of these points from \(P\) and let \(S\) be the closer. Then

1. the farthest point on \(C\) from \(P\) is \(Q\) and the closest such point is \(S\).
2. if \(Q\) is on \(A\) then \(Q\) is the farthest point on \(A\) from \(P\). If \(Q\) is not on \(A\) then the point on \(A\) closest to \(Q\) is the farthest point on \(A\) from \(P\).
3. let \(f\) be a distance function from \(P\) to the points on \(A\), traversed from one endpoint to the other. Then \(f\) is either
   (a) Decreasing with a maximum at the starting endpoint.
   (b) Increasing to a maximum then decreasing.
   (c) Increasing with a maximum at the terminal endpoint.

**Lemma 3** Let \(L\) be a line, \(P\) and \(Q\) two points not on \(L\), and \(Q'\) the reflection of \(Q\) across \(L\). If \(P\) and \(Q\) are on the same side of \(L\), then \(|PQ| < |PQ'|\), otherwise \(|PQ| > |PQ'|\).
We now use these to find the maxpt of $C_{n-4}$, the farthest point on the arcs from $v_{n-4}$. Since one of these arcs is on the same side of $L_{n-2}$ as $v_{n-4}$, all points on this arc can be eliminated by Lemma 3. Now consider the ray $v_{n-4}v_{n-2}$. If this ray intersects the remaining arc then the point of intersection is the maxpt by Lemma 2, with maxspan equal to $|v_{n-4}v_{n-2}| + |v_{n-2}v_n|$. Otherwise the maxpt is an arc endpoint. If the ray passes to the left of the arc (when viewed from $L_{n-2}$), the maxpt is the left-hand endpoint, otherwise the right, again by Lemma 2. The second case is illustrated in Fig. 4. The chain shown is a 4-chain so this is the final step in the algorithm.

![Figure 4](image.png)

Figure 4: The ray $v_0v_2$ passes to the right of the upper arc so the maxpt is $v_4(3,2)$. The maxspan configuration is flat and is achieved by first reflecting $L_3$ and $L_4$ about $L_2$, then $L_4$ about $L_3$. The maxspan is $|v_0v_4(3,2)|$.

To illustrate the first case start with the chain on the left in Fig. 4 but with $L_1$ a bit shorter so that the ray $v_0v_2$ intersects the upper arc. This intersection is the maxpt and the maxspan configuration will occur with $L_3$ and $L_4$ rotated out of $\Pi_1$. See Fig. 5.

![Figure 5](image.png)

Figure 5: The maxspan configuration with links $L_3$ and $L_4$ rotated out of $\Pi_1$. The maxspan is $|v_0v_2| + |v_2v_4|$.

3 The subchain $C_{n-5}$

We have seen that the set of points reachable by $v_n$ under all rotations of the last two links is a partial sphere whose trace in $\Pi_1$ is two arcs symmetric about $L_{n-2}$. We now wish to describe the points reachable by $v_n$ generated by the additional rotation of $L_{n-3}$. Referring again to Fig. 2 the partial sphere consists of a set of circles orthogonal to $\Pi_1$ centered on $L_{n-2}$ extended. When rotated about $L_{n-3}$ each of these circles will generate another partial sphere whose trace on $\Pi_1$ is again two arcs, this time symmetric about $L_{n-3}$. The result is an envelope of circular arcs. We will refer to this set of points on $\Pi_1$ as $R_{n-3}$. Some of the arcs in this envelope are shown in Fig. 6.

![Figure 6](image.png)

Figure 6: The envelope of arcs generated by the rotation of $L_{n-2}$ about $L_{n-3}$.

The shape of these arc envelopes is determined by repeatedly applying Observation 1. If $P$ and $P'$ are on the original two arcs and symmetric about $L_{n-2}$, then the arcs generated by the circle containing these points depend on whether $P$ and $P'$ are on the same or different sides of $L_{n-3}$. For the chain in Fig. 6 all points on both of the original two arcs are on the same side of $L_{n-3}$ extended. When this is not the case $R_{n-3}$ can take on different appearances as in Fig. 7 below.

![Figure 7](image.png)

Figure 7: The boundary of an envelope of arcs when $L_{n-3}$ extended intersects one of the original arcs. $P$ and $P'$ lie on the same side of $L_{n-3}$ so are on the same arc. $Q$ and $Q'$ do not so $Q$ and $Q'$ are on the same arc.

Regardless of appearance in every case $R_{n-3}$ has the following properties:

1. $R_{n-3}$ is symmetric about $L_{n-3}$ extended.
2. $R_{n-3}$ consists of two regions on each side of $L_{n-3}$, each closed and bounded by circular arcs.
3. Each of the eight points in $\Pi_1$ reachable by $v_n$ is an endpoint for an arc on the boundary of $R_{n-3}$.

Looking ahead we observe that when $L_{n-4}$ is allowed to rotate it will result in $R_{n-4}$, the “envelope of an envelope” of arcs, with $R_{n-3} \subset R_{n-4}$. As the complexity
of $R_{n-k}$ increases with each subsequent link rotation we need a way to keep our search for the maxpt as simple as possible. The next result is dedicated to this purpose.

### 3.1 Trimming

In this section we characterize the points in $R_{n-3}$, $n \geq 5$, that could possibly be a maxpt. For this purpose we define the “outer” boundary arcs (or arc portions) of $R_{n-3}$. Let $T$ be a line orthogonal to $L_{n-3}$ which intersects $R_{n-3}$. By symmetry there exist two points on $T$ in $R_{n-3}$, one on each side, farthest from $L_{n-3}$. The set of all such points form the outer boundary arcs which we denote $O_{n-3}$. The arcs in $O_{n-3}$ are shown in bold colors in Fig. 8. The next theorem allows us to exclude all points in $R_{n-3}$ except those in $O_{n-3}$ from further consideration.

**Theorem 4** Only those points on $O_{n-3}$ can be a maxpt for $n$-chains with $n \geq 5$. Furthermore, only these points can generate arcs via subsequent link rotations that could possibly contain a maxpt.

**Proof.** Let $Q$ be a point on the interior of $R_{n-3}$. Then there exists a circle $C$ centered at $Q$ in $R_{n-3}$ as well. Since the line $v_0Q$ intersects $C$ at two points, one of which is farther from $v_0$ than $Q$, $Q$ is not a maxpt. Now let $Q'$ be the reflection of $Q$ about $L_{n-3}$, $Q''$ the reflection of $Q'$ about $L_{n-4}$, and $D$ the disk bounded by $C$. The link rotation about $L_{n-4}$ will generate arcs with endpoints $Q$ and $Q'$ or $Q''$ in $R_{n-4}$. Arcs will be generated for all points in $D$ in a similar manner. So if $A$ is a point on one of these arcs there will exist a disk centered at $A$ which consists of points belonging to the corresponding arcs with endpoints in $D$. Therefore $A$ will be an interior point of $R_{n-4}$ and can therefore not be a maxpt by the preceding argument.

Now let $Q$ be a point on the boundary of $R_{n-3}$ but not in $O_{n-3}$. If $v_0$ is on the same side of $L_{n-3}$ as $Q$, then $Q$ cannot be a maxpt by Lemma 3. Otherwise let $T$ be the line orthogonal to $L_{n-3}$ that contains $Q$. Then $T$ also contains a point $S$ in $O_{n-3}$ farther from $L_{n-3}$ than $Q$. Now $|v_0Q| < |v_0S|$ by the triangle inequality and $Q$ is not a maxpt. So only points in $O_{n-3}$ can be a maxpt. Finally again let $A$ be a point on an arc with endpoints in $D$ as above. Then a line through $A$ perpendicular to $L_{n-4}$ will contain a point in $R_{n-4}$ farther from $v_0$ than $A$ as in the preceding argument and $A$ can not be a maxpt.

These arguments generalize immediately to the outer boundary arcs of $R_{n-k}$ for $3 \leq k \leq n - 1$. So in each iteration we can confine our search for the maxpt to points on these outer boundary arcs.

The process for finding the endpoints of the outer boundary arcs is illustrated in Fig. 8. If a line through the center of a boundary arc parallel to $L_{n-3}$ intersects the arc, then this intersection becomes a new arc endpoint. Portions of the arc below this line are excluded as are any arcs completely below such a line. We refer to this process as trimming the boundary arcs.

**Figure 8.** The only points in $R_{n-3}$ which can be a maxpt for any chain containing $C_{n-5}$ lie on the outer boundary arcs $O_{n-3}$.

### 3.2 Finding the maxpt on $O_{n-3}$

We now turn our attention to the task of locating the maxpt on the set of trimmed boundary arcs $O_{n-3}$. This is simplified by a result which, for $k = 3$ is a direct consequence of Lemma 2. A general proof for arbitrary $k$ is omitted due to lack of space.

**Theorem 5** Let $O_{n-3}$ be the set of outer boundary arcs described above and let $P$ be a point in $\Pi_1$. Define a distance function $f$ from $P$ to the points on $O_{n-3}$ (on the side of $L_{n-3}$ opposite $P$), traversed from one endpoint to the other. Then $f$ is either

1. Decreasing with a maximum at the starting endpoint.
2. Increasing to a maximum then decreasing.
3. Increasing with a maximum at the terminal endpoint.

This result is used to create a simple algorithm for locating the farthest point on $O_{n-3}$ from any point $P$ in $\Pi_1$. If the farthest point from $P$ on any arc $A$ in $O_{n-3}$ is on the interior of $A$ then this is the farthest point from $P$ in $O_{n-3}$. If the farthest point from $P$ for two consecutive arcs is a shared endpoint then this is the farthest point from $P$ in $O_{n-3}$. Otherwise the farthest such point is the starting or terminal arc endpoint in $O_{n-3}$.

As discussed in Section 2 the farthest point from $P$ on each arc $A$ can be determined by drawing a ray from $P$ through the center of $A$. If the ray intersects $A$ then this intersection is the point farthest from $P$. Otherwise
the arc endpoint closest to the ray is the point farthest from \( P \).

Algorithm 1. Maxpt Algorithm

**Input:** A point \( P \) in \( \Pi_1 \) and the set of connected arcs \( O_{n-k} \), \( 2 \leq k \leq n - 1 \), including arc centers, endpoints and radii.

**Output:** The point \( M \) in \( O_{n-k} \) farthest from \( P \).

1. If the farthest point on Arc 1 from \( P \) is the LH endpoint then stop. This is \( M \).
2. If the farthest point on Arc 1 from \( P \) is on the interior of the arc then stop. This is \( M \).
3. Go to Step 1 and repeat with the next arc. If there are no more arcs then the RH endpoint of the last arc is \( M \).

As an example we use the algorithm to find the maxpt of the 5-chain shown in Fig. 9. We work with the arcs on the side of \( L_2 \) opposite \( v_0 \). Begin with the leftmost arc as seen from \( L_2 \). Its center is \( v_3 \) and the closest point to ray \( v_0v_3 \) on this arc is the RH endpoint \( v_5 \). The center of the next (red) arc is \( v_2 \) and the closest point to ray \( v_0v_2 \) on this arc is the LH endpoint \( v_5 \). Thus \( v_5 \) is the maxpt \( M \).

![Figure 9: The farthest point from \( v_0 \) on Arc 1 is the RH endpoint \( v_5 \). The farthest such point on Arc 2 is the LH endpoint, also \( v_5 \). So \( v_5 \) is the maxpt and the maxspan is \( |v_0v_5| \) achieved in the trans-configuration.](image)

We are now in a position to describe our general algorithm. Start with \( O_{n-2} \), the arcs from the rotations of \( L_n \) and \( L_{n-1} \). In each iteration the rotation of link \( L_{i+1} \) about \( L_i \) creates an envelope from which we find the new set of outer boundary arcs \( O_i \). Continue until \( O_2 \) is found. The farthest point \( M \) on \( O_2 \) from \( v_0 \) is the maxpt and \( |v_0M| \) is the maxspan.

4 Boundary Arc Creation

There is one part of this process which has not yet been well described. The question is how to determine the new set of outer boundary arcs \( O_i \) from those in \( O_{i+1} \).

Assume that \( O_{i+1} \) is known and we wish to find \( O_i \). The situation is like that shown in Fig. 6 except that there are now multiple connected arcs symmetric about \( L_{i+1} \) instead of just one. Each symmetric pair of arcs is the trace of a partial sphere. Each pair then generates its own arc envelope when \( L_{i+1} \) is rotated about \( L_i \). The outer boundary arcs of this union of envelopes is \( O_i \) which can be found via the following algorithm. Its justification is given in the Appendix.

Algorithm 2. Boundary Arc Creation Algorithm

**Input:** \( O_{i+1} \), the set of trimmed boundary arcs (endpoints, centers and radii) symmetric about \( L_{i+1} \), and \( v_i \).

**Output:** The trimmed boundary arcs \( O_i \).

1. Use the Maxpt Algorithm to find \( M_{i+1} \) and \( M'_{i+1} \), the points on \( O_{i+1} \) farthest from \( v_i \).
2. **Case 1:** \( M_{i+1} \) and \( M'_{i+1} \) are on the same side of \( L_i \). The arc centered at \( v_i \) with \( M_{i+1} \) and \( M'_{i+1} \) as endpoints is on \( O_i \). Arcs or arc portions between this new arc and \( L_i \) are deleted, all other arcs are kept. Trim the remaining arcs on each side of this new arc with respect to \( L_i \), then reflect all about \( L_i \). This collection of arcs is \( O_i \).
3. **Case 2:** \( M_{i+1} \) and \( M'_{i+1} \) are on opposite sides of \( L_i \). Reflect \( M'_{i+1} \) across \( L_i \) and call this point \( M''_{i+1} \). The arc centered at \( v_i \) with endpoints \( M_{i+1} \) and \( M''_{i+1} \) is on \( O_i \). If \( M''_{i+1} \) is to the left of \( M_{i+1} \) as seen from \( L_{i+1} \) then reflect all arcs and arc portions to the left of \( M_{i+1} \) on \( O_{i+1} \) first across \( L_{i+1} \), then \( L_i \). These reflected arcs belong to \( O_i \) as do those to the right of \( M_{i+1} \). If \( M''_{i+1} \) is to the right of \( M_{i+1} \) then the process is identical except with arcs to the right of \( M_{i+1} \). Trim the remaining arcs on each side of the new arc with respect to \( L_i \), then reflect all about \( L_i \). This collection of arcs is \( O_i \).

In each case only one “new” boundary arc in \( O_i \) is created on each side of \( L_i \). It is the arc of largest radius in the entire envelope. The remaining arcs were either already on \( O_{i+1} \) or are their reflections from the other side of \( L_{i+1} \) about \( L_i \). Case 2 of this algorithm is illustrated in Fig. 10.

5 The Maxspan Algorithm

We now give the entire algorithm.

Algorithm 3. Maxspan Algorithm

**Input:** A chain \( C = (v_0, v_1, \ldots, v_n) \) in flat zigzag configuration.
Output: The maximum span of $C$ expressed in the form $|v_0 v_i(1)| + |v_i(1) v_j(1)| + \cdots + |v_k(1) v_n(1)|$.

1. Initialize. Find $O_{n-2}$. Record the center, radius, and endpoints. Let $i = n - 3$.

2. Find $O_i$, the new (trimmed) outer boundary arcs. Use the boundary arc creation algorithm. Record their centers, radii, and endpoints. Decrease $i$.

3. If $i > 2$ Go to Step 2.

4. Find $M$, the farthest point on $O_2$ from $v_0$. Use the maxpt algorithm. This is the maxpt of $C$.

5. Find the maxspan. If $M$ is an arc endpoint $v_0(1)$ then the maxspan is $|v_0 v_i(1)|$ and the maxspan configuration is flat. Otherwise $M$ is on an arc centered at $v_i$ and the maxspan is $|v_0 v_i(1)|$ plus the radius of this arc. The maxspan configuration will have one or more planar sections rotated out of $\Pi_1$.

6 Computational Complexity

The operations fundamental to each step of the algorithm (reflecting a point about a line, determining if a ray intersects an arc, etc.) are all constant time operations. The complexity of each step is then strictly a function of the number of boundary arcs in each iteration, so as steps are repeated the complexity of the algorithm as a whole depends on the rate of growth of the number of boundary arcs. This is difficult to determine in general since the number of boundary arcs may increase or decrease in each iteration. The number of arcs on each side of $O_i$ may be one more than double the number in $O_{i+1}$ but may also be reduced to just one.

For $\alpha$-chains the number of boundary arcs increases linearly. We sketch the proof as follows: In Case 2 of the boundary arc creation algorithm the maximum number of new boundary arcs per iteration (prior to trimming) is two for all chains, not just $\alpha$-chains. Generally in Case 1 the number of new boundary arcs in $O_i$ can be double plus one the number in $O_{i+1}$. However these arcs are symmetric about $L_{i+1}$ and all remaining vertices $v_0, v_1, \ldots, v_{i-1}$ are on the same side of $L_{i+1}$. So the arcs on the same side of $L_{i+1}$ as the remaining vertices cannot contain $M$ by Lemma 3 and can therefore be trimmed. In this case at most one new arc is added in each iteration. This gives a total of $k \sum_{i=1}^{n} \sum_{j=1}^{n} O(1) = O(n^2)$ operations for any $\alpha$-chain. More generally the number of boundary arcs can increase exponentially until trimming is required in some iteration of the boundary arc algorithm, after which the growth rate tends to be linear. For any given $n$ it is possible to create an $n$-chain $C$ with exponential boundary arc growth, though this can only be done with link lengths that grow exponentially, fixed angles approaching $180^\circ$, or both. These would not likely be present in any modeling environment. As links are repeatedly added to any given subchain trimming will eventually occur. So as $n \to \infty$ the complexity tends to $O(n^2)$.

References