# Approximation of an Open Polygonal Curve with a Minimum Number of Circular Arcs and Biarcs

R. L. Scot Drysdale [a] Günter Rote [b,1] Astrid Sturm [b,1]

[a] *Department of Computer Science, Dartmouth College*

[b] *Institut für Informatik, Freie Universität Berlin, Takustr. 9, 14195 Berlin, Germany*

**Abstract**

We present an algorithm for approximating a given open polygonal curve with a minimum number of circular arcs. In computer-aided manufacturing environments, the paths of cutting tools are usually described with circular arcs and straight line segments. Greedy algorithms for approximating a polygonal curve with curves of higher order can be found in the literature. Without theoretical bounds it is difficult to say anything about the quality of these algorithms. We present an algorithm which finds a series of circular arcs that approximate the polygonal curve while remaining within a given tolerance region. This series contains the minimum number of arcs of any such series. Our algorithm takes $O(n^2 \log n)$ time for an original polygonal chain with $n$ vertices. Using a similar approach, we design an algorithm with a runtime of $O(n^2 \log n)$, for computing a tangent-continuous approximation with the minimum number of biarcs, for a sequence of points with given tangent directions.

## 1 Introduction

In computer-aided manufacturing environments, tool paths are usually made of line segments and circular arcs [11–13]. Approximating the data by curves of

*Email addresses:* scot@cs.dartmouth.edu (R. L. Scot Drysdale), rote@inf.fu-berlin.de (Günter Rote), sturm@inf.fu-berlin.de (Astrid Sturm).

[1] Partially supported by the IST Programme of the EU as a Shared-cost RTD (FET Open) Project under Contract No. IST-006413 (ACS – Algorithms for Complex Shapes)

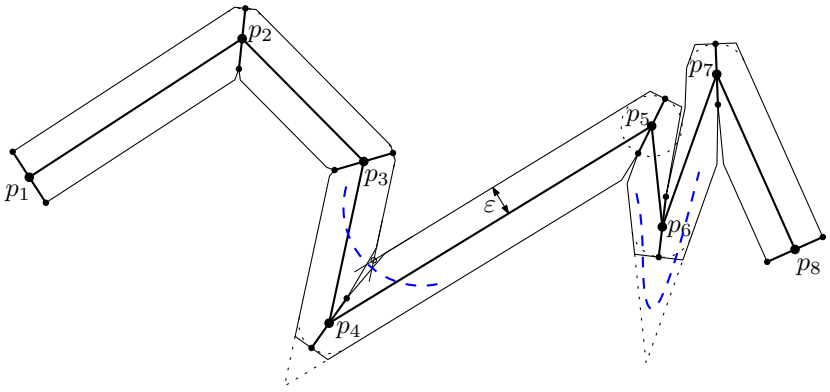


Fig. 1. Polygonal tolerance region $R$ with gates

higher order [5,8,11–13,15,16,20] has been investigated extensively in the past. In contrast to approximation by *polygonal* curves, the theoretical bounds of these problems are not so well studied. There are two types of optimization problems associated with the polygon approximation problem:

- Min-# problem: Given $\varepsilon \geq 0$, construct an approximate curve with "error" within $\varepsilon$ and having the minimum number of line segments.
- Min-$\varepsilon$ problem: Given $m$, construct an approximate curve consisting of at most $m$ line segments with minimum approximation "error".

As these optimization problems were answered for polygonal approximation [2,7,9,10,14,18,19] the same questions arise for approximation with curves of higher order. We were able to answer the min-# problem for approximating open polygonal curves with circular arcs and for biarcs.

In the first part of this paper we will introduce the basic ideas and the algorithm for approximation of a polygonal curve with the minimum number of circular arcs. Building on this we later present an algorithm for tangent-continuous approximation of an open polygonal curve with minimum number of biarcs.

### *1.1 Results and Techniques*

We assume that we are given a *tolerance region* around the given curve, which is split into subregions by *gates* through the given points, see Figure 1. The precise formulation is given below.

Our algorithm for the optimal approximation by circular arcs (Section 2) determines a subsequence of the input vertices and connects them by a sequence of circular arcs, lying in the tolerance region and intersecting the gates in proper order, thereby remaining close to the input polygon chain. The algo-

rithm finds the approximation with the minimum number of arcs, subject to these constraints.

The main idea for this algorithm is the use of a Voronoi diagram of the tolerance boundary. We have to incrementally maintain one cell in this Voronoi diagram of line segments. Geometric considerations (Lemma 2.9) make the location step in the update easy, leading to constant amortized time per insertion. In total, the algorithm takes $O(n^2 \log n)$ time and $O(n)$ space.

We also obtain an optimal tangent-continuous approximation with *biarcs*, pieces consisting of pairs of circular arcs, with given tangent directions in $O(n^2 \log n)$ time and $O(n^2)$ space (Section 3). (If such tangent information is not available, it can be computed from the point data alone, using various tangent estimation methods.) The algorithm selects a subsequence of the input vertices and connects them by biarcs, respecting the tangent directions of the chosen vertices. Again, the approximation remains within the tolerance region. The resulting approximation uses the minimum possible # of biarcs, subject to these constraints.

Conventional biarc algorithms (also used in industry) operate on discrete sets of points (and tangent vectors), by fitting biarcs between selected pairs of points [8]. Therefore the restrictions of our algorithms are common. Nevertheless, we are aware that certain restrictions of the solution are not completely natural. In particular, one might allow arcs and biarcs that do not start and end at original points. Using these restrictions simplifies the problems, and we do not know to solve them otherwise.

The results on arc approximations in Section 2 have been presented at the 22nd European Workshop on Computational Geometry (EWCG) in Delphi, in March 2006 [4].

## 2 Approximation by Circular Arcs

### *2.1 Problem Setting*

We wish to approximate a polygonal chain $P = (p_1, \ldots, p_n)$ by a series of circular arcs (which could include straight line segments, as the limiting case of circles of infinite radius). The endpoints of the arcs are vertices of $P$. Ideally, we want our approximating curve to have distance at most $\varepsilon$ from $P$. As a first approximation to this problem, one can look at a region formed from strips of width $\varepsilon$ centered at the polygon edges. However, in the vicinity of sharp corners, this does not guarantee that the curve remains close to the

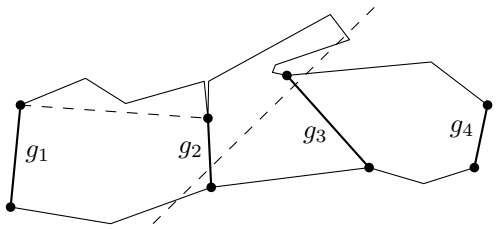


Fig. 2. Polygonal tolerance region $R$ with gates. Gate $g_2$ has been "shortened" to fulfill condition B between $g_1$ and $g_2$. Condition C is violated between $g_2$ and $g_3$.

given points. Figure 1 shows a circular piece of a hypothetic curve that can shortcut the bend at $p_4$ if it is only required to remain in the strips. (Also, it might overshoot the bend, as indicated in the vicinity of $p_6$, although this looks like a theoretical possibility only.) To avoid this, we introduce a *gate* through every vertex. The approximating curve is required to pass through all gates in succession, and the curves are not allowed to pass through a gate twice. This will guarantee that any curve into a point $p_i$ can be joined with any curve out of $p_i$ without danger of an intersection other than at $p_i$.

For our problem, we assume that we are given a polygonal "tolerance region" $R$ and a sequence of gates $g_1, g_2, \ldots, g_n$, which are segments through the points $p_i$. Each gate crosses $P$. We will refer to endpoints of gates lying to the left of $P$ as we walk from $p_1$ to $p_n$ as left endpoints and the other endpoints as right endpoints. We require that the gates do not cross each other. We require that the input satisfies the following assumptions:

(A) $R$ is a simple polygon passing through all gate endpoints; the boundary of $R$ goes through $g_1$ and $g_n$.
(B) $R$ does not intersect the interior of gates or cross the segments connecting corresponding endpoints of successive gates.
(C) No line through two points on successive gates $g_i$ and $g_{i+1}$ crosses the portion of $R$ connecting $g_i$ with $g_{i+1}$.

(Assumption (B) is actually a consequence of (C).) Ideally, the gate $g_i$ at vertex $p_i$ is a line segment of length $2\varepsilon$ centered at $p_i$ that bisects the angle $p_{i-1}p_ip_{i+1}$. For a convolved curve with sharp bends close together, we might have to reduce the width of $R$ in order to fulfill condition A; and we might have to shorten the gates in order to fulfill condition B, as shown in the right part of Figure 1 and in the left part of Figure 2. In contrast, condition C, beyond what is required for condition B, is likely not an issue in practice: it prevents the boundary of $R$ from making "wild" turns like in the middle of Figure 2. The end gates $g_1$ and $g_n$ partition the boundary of $R$ into a *left* boundary and a *right* boundary. In the illustrations, $P$ will usually be oriented from left to right; then the left boundary is on top and the right boundary is below.

Modeling the curve approximation problem by an appropriate tolerance re-

gion with gates is a problem of its own, which we do not treat here. Eibl and Held [5,8] have methods that can be adapted to produce such gates and tolerance regions. In Figure 1, we have chosen to approximate the "ideal" circular boundary at the outer angle of each vertex by a single edge of $R$. One can use more edges to get a finer approximation, or one could also choose to approximate the circular arc from inside, to get a guaranteed upper distance bound of $\varepsilon$. Our time bounds assume that $R$ has constant complexity between successive gates and thus the total size of $R$ is proportional to $n$. See Section 4 for a discussion of further issues about the tolerance region.

**Definition 2.1 (proper gate stabbing)** *A circular arc stabs gates $g_i, g_{i+1}, \ldots, g_j$ properly, if:*

*(1) the circular arc passes through each gate $g_m \in \{g_i, \ldots, g_j\}$ from the side of segment $\overline{p_{m-1}p_m}$ to the side of segment $\overline{p_m p_{m+1}}$*
*(2) the circle on which the arc lies intersects each gate only once.*

Condition 2 of this definition is necessary for our algorithm, but it excludes arcs that might seem reasonable: an arc from $p_i$ to $p_j$ might intersect each intermediate gate only once, but the continuation of the arc beyond $p_j$ might bend back and intersect, say, $g_j$ and $g_{j-1}$ a second time, see Figure 3. This would be a sensible arc, but it is excluded by our definition. But such a situation can only happen if the gates are very close together (relative to their length).

**Definition 2.2 (valid circular arc)** *A circular arc $a_{ij}$ with starting point $p_i$ and endpoint $p_j$ is a valid arc if:*

- *the arc stabs the gates $g_{i+1}, \ldots, g_{j-1}$ properly,*
- *the arc does not cross the boundary of the tolerance region $R$.*
- *the arc reaches $p_i$ from the correct side of $g_i$ and reaches $p_j$ from the correct side of $g_j$.*

Note that because $R$ passes through the gate endpoints, any arc that goes through a series of gates without crossing the tolerance boundary must go through them in the correct order, so we do not need to test this separately. In contrast to the intermediate gates, we allow the circle on which the arcs lie to intersect $g_i$ and $g_j$ more than once.

We can split the problem of determining if a valid circular arc connects $p_i$ with $p_j$ into three parts. First, we compute the set of all arcs between $p_i$ and $p_j$ that stab all intermediate gates properly (Sect. 2.2). Second, we compute all arcs that start at $p_i$ and end at $p_j$, reaching both from the correct side (Sect. 2.3). Third, we compute all arcs between $p_i$ and $p_j$ that do not intersect with the tolerance boundary (Sect. 2.4). A valid circular arc has to be a member of all three result sets.

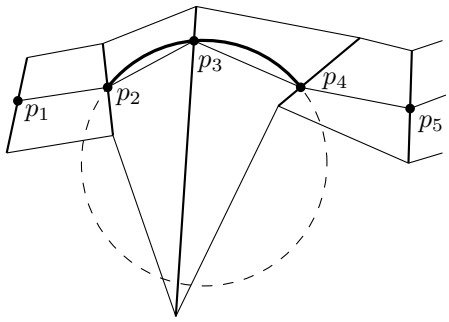


Fig. 3. A circular arc in a hypothetical tolerance region $R$ that is not valid because it violates Condition 2 of Definition 2.1.

*2.2 Stabbing the Gates*

Given a point $p$ and a gate $g$, denote by $b_l$ the bisector of $p$ and $g$'s left endpoint, and by $b_r$ the bisector of $p$ and $g$'s right endpoint.

**Lemma 2.3** *The centers of all circles passing through a vertex $p_i$ and intersecting a gate $g_j$ exactly once lie in a double-wedge whose boundary is $b_l$ and $b_r$. Specifically, they lie in the parts of the double wedge where one of the half planes bounded by $b_l$ and $b_r$ includes $p_i$ and the other excludes it. (Figure 4 illustrates this.) In the degenerate case where $b_l$ is parallel to $b_r$ the region containing all centers is the strip between the bisectors.*

**PROOF.** Consider the intersection of the half plane bounded by $b_l$ that excludes $p_i$ and the half plane bounded by $b_r$ that includes $p_i$. Points in the interior of this region are closer to $p_i$ than the right endpoint of the gate and are also closer to the left endpoint than to $p_i$. Disks centered in this region which have $p_i$ on their boundary include the left endpoint and exclude the right endpoint of the gate. Therefore all circles centered in the wedge intersect the gate exactly once. The case for the second wedge is symmetric. This argument works for the degenerate case, also, but in this case all circles will include the nearer gate endpoint and exclude the further one.

Centers of circles that are located in the same region as $p_i$ outside of the double-wedge are always closer to $p_i$ than to the endpoints of the gate. Therefore these circles exclude the endpoints if they pass through $p_i$. These circles can not intersect the gate only once, unless the circle is tangent to the gate. Looking at the other side of the double-wedge boundary, all centers of circles located here are closer to the endpoints of the gate than to $p_i$. Each disk which includes $p_i$ has to include the endpoints and its boundary does not intersect the gate at all. □

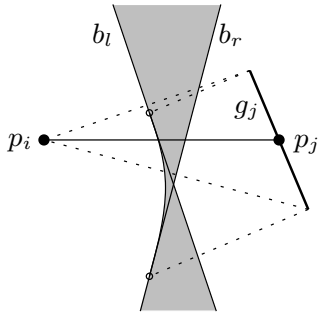


Fig. 4. The shaded area is the region of all centers of circles passing through $p_i$ and gate $g_j$. The circles with centers close to the intersection of $b_l$ and $b_r$, in the region with the curved boundary, intersect $g_j$ twice and are not considered as centers of valid arcs.

By Definition 2.1, an arc stabs the gates properly only if every gate is intersected only once. Therefore the centers of circular arcs stabbing an intermediate gate are located in the double wedge of the gate. For the first and last gates of the arc we insist that the arc goes through the original point located at the gate. Thus the first and last gates are treated differently from the intermediate gates (see Subsection 2.3).

According to Lemma 2.3, one wedge is the region of the centers of disks including the left endpoint of the gate and excluding the right endpoint. Circular arcs centered in this region pass the gate from the correct side, according to the stabbing condition, if they are in CCW (counter-clockwise) orientation. In CW (clockwise) orientation, the arc would walk around the left endpoint before intersecting the gate. The unbounded part of this wedge lies to the left of $P$. Symmetrically the circular arcs in the other wedge need CW orientation to pass the gate in the correct direction, and the unbounded part of this wedge lies to the right of $P$.

So from now on we talk about the left wedge and the right wedge. A circular arc stabbing through the gates cannot change its orientation.

**Lemma 2.4** *A circular arc $\alpha$ starting at a point $p$ stabs gates $g_i, \ldots, g_j$ properly if and only if its center lies in the intersection of the left wedges or the intersection of the right wedges defined by $p$ and the gates.* □

**Lemma 2.5** *Incrementally computing the two regions of centers of all valid circular arcs passing through a point $p_i$ and stabbing all gates $g_i, g_{i+1}, g_{i+2} \ldots, g_j$ properly, for $j = i+1, \ldots, n$, can be done in $O(n \log n)$ time and $O(n)$ space.*

**PROOF.** It is the incremental intersection of $O(n)$ half-planes. □

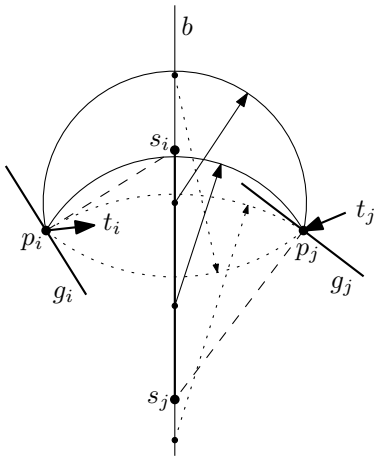


Fig. 5. Illustration for Lemma 2.6. In this example, arrows labeled $t_i$ and $t_j$ show the direction that $P$ passes through gates $g_i$ and $g_j$. The dashed segments perpendicular to the gates at $p_i$ and $p_j$ show how $s_i$ and $s_j$ are computed. The centers of valid CW arcs form the line segment $s_i s_j$. There are no valid CCW arcs. A few representative candidate arcs are shown. The two solid arcs (with solid radii) centered at points between $s_i$ and $s_j$ are valid. The two dotted arcs (with dotted radii) centered at points outside of segment $s_i s_j$ are invalid.

### *2.3 Arc Endpoints*

All arcs that start at $p_i$ and end at $p_j$ have their centers on the bisector of the segment connecting $p_i$ and $p_j$. Since a valid circular arc from $p_i$ to $p_j$ must reach each endpoint from the correct side of its gate, we know for each circle whether the arc from $p_i$ to $p_j$ must go in the CW or in the CCW direction, or if none of the arcs is valid. (When the circle is tangent to both gates, both directions are possible.) Straightforward geometric arguments lead to the following characterization of the desired arcs, see Figure 5.

**Lemma 2.6** *Let $b$ be the perpendicular bisector of the segment between $p_i$ and $p_j$. Let $s_i$ be the point of $b$ which is the center of a circle tangent to $g_i$ at $p_i$, and let $s_j$ be defined symmetrically. The centers of all CW arcs that reach both $p_i$ and $p_j$ from the correct side lie in the intersection of two rays that are subsets of $b$. One has $s_i$ as its endpoint and the other has $s_j$ as its endpoint. The same is true for CCW arcs.* □

### *2.4 Staying within the Tolerance Boundary*

The tolerance boundary $R$ consists of two polygonal chains, one on each side of the original polygonal chain $P$. For a CW arc we will only check that it does not cross the boundary on the *left* side of $P$. It cannot cross the boundary on the right side of $P$ if it passes through all gates, by assumption (B), and

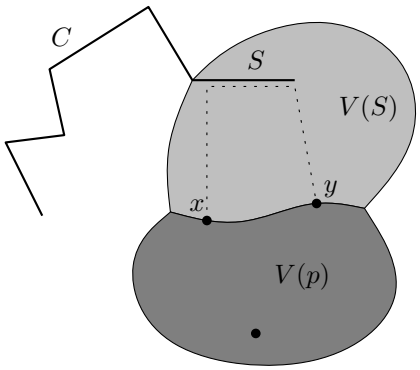


Fig. 6. Schematic illustration for Lemma 2.8.

therefore we need not check for such an intersection explicitly. (For a CCW arc, the situation is symmetric.)

A circle passing through point $p$ does not intersect or contain any edge on a polygonal chain $C$ if its center lies closer to $p$ than to any point on $C$. That is, if we compute the Voronoi diagram of $C \cup p$, the center of the circle must lie in point $p$'s region, $V(p)$.

This is not quite the condition that we want, namely that a circular arc does not cross chain $C$. The Voronoi region guarantees that an entire circle does not cross $C$. However, in our case these are equivalent.

**Lemma 2.7** *If an arc from $g_i$ to $g_j$ does not intersect a tolerance boundary between $g_i$ and $g_j$ then neither does the circle on which that arc lies.*

**PROOF.** Look at the arc between consecutive gates $g_k$ and $g_{k+1}$. Let $q$ and $q'$ be the intersection points with these gates. By assumption (C), the line $\ell$ through $q$ and $q'$ does not intersect the tolerance boundary between $g_k$ and $g_{k+1}$, i.e., the tolerance boundary lies entirely on one side of $\ell$. For a CW arc, the tolerance boundary in question lies on the left side of $\ell$. On the other hand, $\ell$ is the line that splits the circle into the arc from $q$ to $q'$ (on the left side) and into the opposite part which is not used. Thus the part of the circle which is not used can never intersect the relevant part of the tolerance boundary. □

While we could compute the entire Voronoi diagram of $C \cup p$ to determine $V(p)$, this would be too expensive. Fortunately, we can iteratively add $n$ consecutive segments of $C$ and update $p$'s Voronoi region $V(p)$ in $O(n)$ total time.

Voronoi regions are "generalized star shaped". This means that a shortest segment from a boundary point to a nearest point in the shape defining the region lies entirely within the region.

**Lemma 2.8** *Each segment added will either cause no change to $V(p)$ or will replace a section of $V(p)$ by at most three new segments (two straight lines and a parabola). (If $V(p)$ is unbounded we think of an edge "at infinity" connecting the two infinite rays, so that these three "segments" are considered consecutive.)*

**PROOF.** Suppose we add a new segment $S$ to the end of $C$. First we show that the added pieces on the boundary of $V(p)$ are connected. Let $x$ and $y$ be two points on the boundary between $V(p)$ and $V(S)$ ($x$ and $y$ can also be chosen "at infinity"). Draw shortest segments (or rays for the piece "at infinity") from $x$ and $y$ to $S$. Because Voronoi regions are generalized star shaped, both of these segments lie within $V(S)$ and cannot be crossed by another Voronoi region. $S$ itself cannot be crossed by another Voronoi region. There is a closed curve formed by a part of $S$, the two segments, and the boundary of $V(p)$ between $x$ and $y$, cutting the plane into two parts, see Figure 6. Since $S$ is the (current) last segment of $C$, one of these parts contains no other segments of $C$. It follows that the corresponding part on the boundary of $V(p)$ belongs completely to $V(S)$, establishing a connection between $x$ and $y$.

The Voronoi bisector between a point $p$ and a segment $S$ is formed by 2 straight rays and a parabolic arc. The new parts on the boundary of $V(p)$ must be a part of this bisector. □

There are two parts to updating $p$'s Voronoi region $V(p)$ when adding a segment $S$ to the diagram. First, we find a place on the boundary of $V(p)$ that is equidistant from $p$ and $S$, if such a place exists. If so, we walk around the boundary of $V(p)$, eliminating boundary sections until we reach the other place on the boundary where $p$ is equidistant from $S$. (Note that either of these places could be "at infinity".)

The second part is easy — walk around the boundary of $V(p)$ from the starting point, eliminating obsolete bisector segments until you get to the finish point.

Because $C$ is a polygonal chain, the first part is also easy. $V(p)$ is bounded by bisector pieces between $p$ and a subset of the segments in $C$. Of the segments in this subset, there is a first segment $F$ and a last segment $L$, according to the order along the chain.

**Lemma 2.9** *If $V(p)$ changes, then its boundary with either $V(F)$ or $V(L)$ must change.*

**PROOF.** The intuition is, if you can't go through the chain $C$, then the only way to get to $V(p)$ is through $V(F)$ or $V(L)$.

If the chain from $F$ to $L$ consists of only $F$ and $L$ (which could be the same segment), the lemma is trivially true. Otherwise consider the union of the chain $C$ between $F$ and $L$ exclusive, the boundary of $V(F)$ from the endpoint it shares with the next segment on $C$ to the end of its boundary with $V(p)$, and the boundary of $V(L)$ from the endpoint it shares with the segment before it on $C$ until the end of its boundary with $V(p)$. If $V(p)$ is bounded these two boundaries end at the same point — the point where $V(p)$, $V(F)$, and $V(L)$ meet. If $V(p)$ is unbounded then its boundaries with $V(F)$ and $V(L)$ end in infinite rays. In either case, this union separates the plane into two parts, one including $p$ (the inside) and the other not including $p$ (the outside). We will call this union the separator. Note that $F$ and $L$ are defined to lie outside of this separator (except for the endpoint that is part of the separator).

Suppose that a segment $S$ is added that changes $V(p)$. The previous segment on $C$ is either $L$ or some segment that did not modify $V(p)$. In either case, the endpoint shared with that previous segment is outside of the separator, so we know that at least part of $S$ lies outside of separator.

If $S$ crosses the separator, then it cannot cross $C$, because the chain is simple. If it crosses the Voronoi boundary of $V(F)$ then the part of the boundary between the crossing point and the end of the boundary between $V(p)$ and $V(F)$ will be eliminated. A similar argument holds for $L$. Thus if $S$ crosses the separator then the boundary of $V(p)$ with either $V(F)$ or $V(L)$ must change.

If $S$ does not cross the separator, pick some point $q$ that is on the boundary of the new $V(p)$ that was not on the boundary of the old $V(p)$ and let $E$ be the shortest segment from $q$ to a point on $S$. $E$ must lie entirely in $V(S)$ and must cross the separator. It cannot cross $C$. The rest of the analysis is exactly as in the paragraph above, with $E$ replacing $S$. □

**Lemma 2.10** *For a fixed gate $g_i$, we can incrementally compute the regions of centers of all circular arcs that pass between $g_i$ and each gate $g_j$, without crossing the tolerance boundary, for $j = i+1, i+2, \ldots, n$, in $O(n)$ time and space.*

**PROOF.** Incrementally add segments from $C$ and amortize the update time. We have shown that the centers of CW arcs are the region of $V(p_i)$ in the Voronoi diagram of $p$ along with the CW boundary between $g_i$ and $g_j$, and that this statement is true when "CCW" is substituted for "CW". We can compute these regions incrementally. It takes constant time to test if segment $S$ changes the boundary between $p$ and either $F$ or $L$, so the total time for finding starting points is $O(n)$.

Walking along the boundary of $V(p)$ will take time proportional to the number of pieces eliminated. Because an eliminated piece is removed and never

reappears, the total time for this step in all $n$ insertions is bounded by the number of boundary pieces added. This is at most $3n$, because a bisector curve between $p$ and a segment consists of at most three pieces. Thus this requires time $O(n)$. □

### *2.5 Computing the Shortest Path*

To determine the approximation with the minimum number of arcs we look at the directed acyclic graph of all possible valid arcs and find the shortest path from $p_1$ to $p_n$. The following theorem summarizes how to find the valid arcs from $p_i$ to $p_j$.

**Theorem 2.11** *A point $c$ is the center of a valid CW circular arc from $p_i$ to $p_j$ if and only if it is in the intersection of*:

- *the intersection of the right wedges between $p_i$ and each of the gates $g_{i+1}$ through $g_{j-1}$;*
- *the region of $V(p_i)$ in the Voronoi diagram of $p_i$ and all of the segments on the left boundary between $g_i$ and $g_j$; and*
- *all points in the intersection of two rays contained in $b$, one with endpoint $s_i$ and the other with endpoint $s_j$, where $b$, $s_i$, and $s_j$ are as defined in Lemma 2.6.*

*The conditions for valid CCW arcs are symmetric.* □

We find the possible arcs from a point $p_i$ to all points further along $P$ incrementally. We maintain the intersection of the right wedges, the intersection of the left wedges, the Voronoi region of $p_i$ with the left boundary, and the Voronoi region of $p_i$ with the right boundary. At each step we update each of the four items. We intersect each bisector ray with an intersection of wedges and with a Voronoi region, and then test if the intersections overlap. Because wedge intersections and $V(p_i)$ are convex these intersections require $O(\log n)$ time.

Note that we can quit early as soon as both wedge intersection regions become empty. This may lead to a better behavior of the algorithm in practice than the worst-case time bound proved in the theorem below.

**Theorem 2.12** *Given an open polygonal curve $P = (p_1, \ldots, p_n)$, a polygonal tolerance boundary of size $O(n)$, and a gate for each $p_i$, we can approximate $P$ by a minimum number of valid circular arcs in $O(n^2 \log n)$ time and $O(n)$ space.*

**PROOF.** For each starting point $p_i$ we can determine the points $p_j$ $(j > i)$ that can be reached by a valid arc in $O(n \log n)$ time and $O(n)$ space. In the shortest path algorithm, we are performing a breadth first search of the graph, but we need store only the tree edge of the BFS tree. It is therefore sufficient to scan the outgoing arcs of $p_1, p_2, p_3$, and so on, in succession as they are expanded in the BFS. Once the valid arcs out of $p_i$ are scanned, we discard all edges that are not part of the BFS tree, and hence the algorithm needs only $O(n)$ space. □

For the Min-# problem for *polygonal* approximation the best known running time is $O(n^2 \log n)$ for three out of the four common error criteria [6,9,10]. Our algorithm solves this problem with curves of higher order with the same time complexity. The error criteria refer to the way the error of an approximating segment is measured. Only for the error criterion where error is defined to be the maximum distance between the approximating segment and the vertices of the original polygonal curve that lie between start and endpoint of the segment is there is an algorithm with a faster running time of $O(n^2)$ [10].

**Remark.** The algorithm can be extended to optimize other criteria than the number of arcs, e.g. the arc length, or some weighted mixture of criteria. When the interval of possible centers of valid arcs from $p_i$ to $p_j$ has been determined, one must be able to pick the best one of them and compute its "weight", which is used for the shortest path calculation.

## 3 Approximation by Biarcs

### *3.1 Problem Setting*

The sequence of arcs produced in the previous algorithm may have arbitrary corners at the vertices. In many situations, a smooth curve is desired. We now assume that an oriented tangent direction $t_i$ is specified for each vertex $p_i$ of the open polygonal curve. (If such tangent information is not available, it can be computed from the point data alone, using various tangent estimation methods.)

Our algorithm will select a subsequence of the input points and interpolate between them smoothly by *biarcs*, pieces consisting of pairs of circular arcs, respecting the tangent directions $t_i$ at the points which are used. Our algorithm will find such an approximation with the minimum number of biarcs given a set of gates and a tolerance region. In this setting the gates would ideally

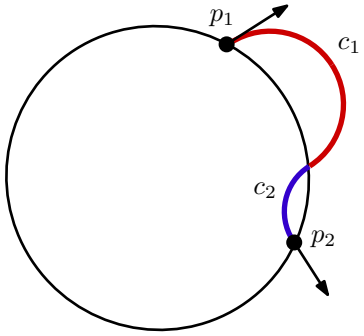


Fig. 7. The joint circle, and an S-shaped biarc with both tangents pointing outside the joint circle

be perpendicular to the tangent directions $t_i$, but we do not require this. We require that the input satisfies the following assumptions:

(A′) $R$ is a simple polygon passing through all gate endpoints;
(B′) $R$ does not intersect the polygon or the interiors of the gates.
(C′) Each tangent $t_i$ passes through gate $g_i$ in the same direction as the original polygonal chain $P$; that is, from the side of the gate on which $p_{i-1}p_i$ lies to the side on which $p_ip_{i+1}$ lies.

Again we first find all valid biarcs and then build the directed graph of these biarcs from the start point to the end point of the polygonal curve. The last step is the computation of the shortest path as in the previous section. The difference between the two algorithms is the computation of the valid arcs/biarcs. Therefore we will now focus on the computation of valid biarcs.

### *3.2 Biarcs*

Biarc curves were introduced by Bolton [1] and are used for curve approximation in a tangent-continuous manner. A biarc consists of two circular arcs that share an endpoint with a common tangent. This common endpoint is called the *joint* of the biarc. Given two points $p_i$ and $p_j$ with two tangent vectors $t_i, t_j$ at these points, a biarc $B_{ij}$ between $p_i$ and $p_j$ is characterized in the following way [1,8]:

- $B_{ij}$ consists of two consecutive circular arcs, $a_1, a_2$
- $a_1$ is an oriented arc from $p_i$ to point $p_{\text{joint}}$ and $a_2$ is an oriented arc from $p_{\text{joint}}$ to $p_j$;
- $a_1$ matches the tangent vector $t_i$ at the point $p_i$ and $a_2$ matches the tangent vector $t_j$ at $p_j$;
- both arcs have a common tangent at $p_{\text{joint}}$.

These conditions leave one degree of freedom. The locus of possible joints forms a circle $J$ that passes through $p_i$ and $p_j$ [3,17,21], see Figure 7. For each point on this *joint circle* $J$, there is a unique biarc which uses this point as the joint. (There are some degenerate cases, which we ignore in the sequel: as a limiting case, the joint could be one of the points $p_i$ or $p_j$: the joint circle might be a line; if there is a circle through $p_i$ and $p_j$ with the given tangents, this is the joint circle, but all joints on this circle lead essentially to the same biarc.)

**Proposition 3.1** *One circular arc of the biarc lies outside the joint circle $J$, and the other lies inside $J$, except for their endpoints, which lie on $J$. Both tangents $t_i$ and $t_j$ point to the same side (either inside or outside) of $J$, and they form equal angles with $J$.*

(In fact, the last property characterizes the joint circle.)

### *3.3 Valid Biarcs*

**Definition 3.2 (Valid biarc)** *A* valid biarc *$B_{ij}$ from $p_i$ to $p_j$ consists of two circular arcs $a_1$ and $a_2$ and satisfies the following conditions*:

- *$a_1$ matches $t_i$ at the point $p_i$, $a_2$ matches $t_j$ at $p_j$, and they meet at a point on the joint circle.*
- *$B_{ij}$ stays inside the tolerance boundary.*
- *$B_{ij}$ intersects the gates $g_i$ and $g_j$ only in $p_i$ and $p_j$.*

The joint, which is the ending point of $a_1$ and the starting point of $a_2$, is not required to be an original point. The joint must of course lie inside the tolerance region. Note that in comparison to the arc approximation of Definition 2.2, we have relaxed the gate stabbing condition. The arcs $a_1$ and $a_2$ are allowed to intersect the gates of the starting and ending points only once, but intermediate gates can be intersected more than once. Forbidding these multiple intersections would mean that, in a family of biarcs with the same endpoints, some biarcs that lie between permitted biarcs might be excluded, which is not natural. See Figure 8. The restrictions on intersecting $g_i$ and $g_j$ guarantee that successive biarcs will not intersect except at endpoints.

### *3.4 Circular Visibility Regions*

For each possible starting point $p_i$ of a biarc, the tangent direction $t_i$ is fixed. The pencil of circular arcs starting in this direction forms a *circular visibility region* $W_i$ inside the feasible region $R$, see Figure 9. The arcs forming $W_i$

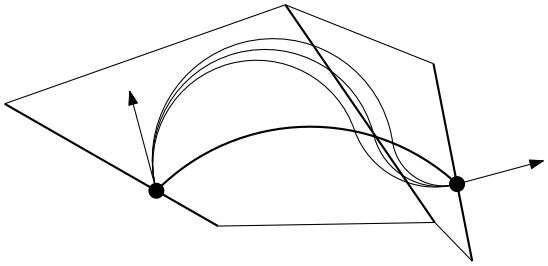

Fig. 8. We allow a biarc to intersect an intermediate gate three times.

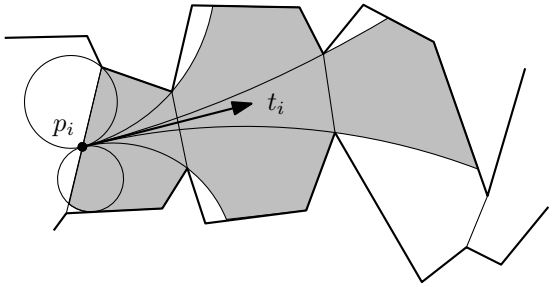


Fig. 9. A circular visibility region $W_i$

terminate when they reach $g_i$; since we want to construct valid biarcs, we are not interested in arcs that intersect $g_i$ a second time.

To find a valid biarc that starts at $p_i$ and ends at $p_j$ we need to reach a point on the joint circle $J$ via a valid arc from $p_i$ and then continue via a valid arc to $p_j$. The possible joints from the perspective of $p_i$ are the intersection of $J$ and the circular visibility region of $p_i$. By reversing the direction of the second arc and tangent we can compute the second arc in the same way. We will use arc $\tilde{a}_2$, which has opposite orientation and whose tangent at $p_j$ is $\tilde{t}_j$, the reverse of $t_j$. We will call this circular visibility region $\tilde{W}_j$. Our goal is to find all points on $J$ which are in both circular visibility regions $W_i$ and $\tilde{W}_j$.

As a first step in this process we determine the portion of each gate that is within $W_i$ for each point $p_i$ and the portion of each gate that is in $\tilde{W}_j$ for each point $p_j$. These portions consist of at most three intervals and can be stored in $O(n^2)$ space (see Lemma 3.3). In the second step we check the existence of a valid biarc between every pair of vertices $p_i$ and $p_j$. For each pair $p_ip_j$, we will identify an interval $g_{l-1}, \ldots, g_{r+1}$ of gates where the joint is restricted to lie. Between $g_l$ and $g_r$, the joint circle is not intersected by the boundary of $R$. This makes it easy to test for the existence of a valid joint. This step uses the pre-computed information about the intersection of circular visibility regions with gates.

We could compute the intersection of $W_i$ with all later gates by using intersections of wedges and Voronoi regions, as we did in Section 2. However, because we know the tangent at $p_i$ we can do this more efficiently by computing $W_i$ directly. The pencil of circular arcs consists of an interval of possible curvatures.

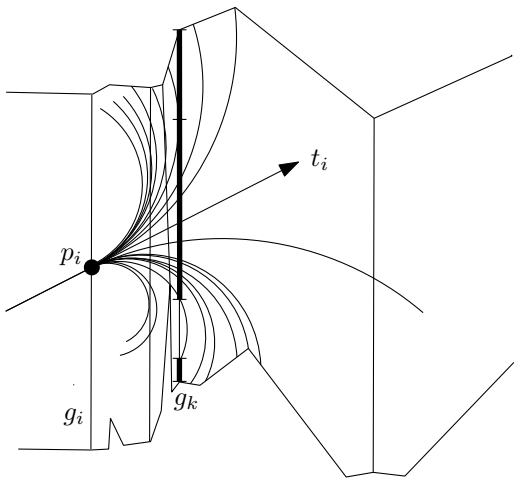


Fig. 10. Forward and backward visibility segments of region $W_i$ on gate $g_k$. The intervals of forward visibility and counter-clockwise backward visibility are adjacent.

As we proceed from gate to gate and walk along the left and right tolerance boundaries, the interval of curvatures either remains unchanged or shrinks, but it always remains a single interval.

**Lemma 3.3** *For a given point $p_i$, the oriented circular visibility regions of $W_i$ and their intersection with all gates can be computed in $O(n)$ time. The part of a gate that is visible in $W_i$ consists of at most three intervals: one interval where the gate is reachable in the forward direction, and two segments where the gate is reachable in the backward direction by clockwise and counter-clockwise arcs, respectively. These intervals may be adjacent.*

**PROOF.** We cut each oriented visibility region into two pieces, forward and backward visibility. The forward visibility region is the part of the visibility region which is reached by portions of arcs that do not intersect any gate twice. The backward visibility is the part reached by portions of arcs after they have intersected a gate twice, so they are moving backwards through the gates.

We walk along the left and right boundaries of the tolerance region, determining the intersection between each boundary and the pencil of arcs, and in this way compute the forward visible region. When the last reachable gate is known, we can compute for each gate moving backwards the arcs that build the backward visibility region. The backward part of the visibility region for a gate $g_i$ consists of the arcs that intersect gate $g_{i+1}$ a second time (possibly after passing through even higher-numbered gates twice) and reach back to $g_i$, plus the arcs that don't cross $g_{i+1}$ but intersect the gate $g_i$ a second time. These arcs correspond to a connected piece of the pencil of arcs and we need to determine the intersection of this pencil part with the corresponding boundary of the tolerance region moving from $g_{i+1}$ to $g_i$. Because the complexity of the

tolerance boundary between two gates is constant we can do the forward and backward visibility computations between two gates in constant time, so the total time required is $O(n)$. The intervals on the gates can be stored for all point pairs in $O(n^2)$ space. $\square$

Note that the interval on a gate reachable by forward portions of arcs is disjoint from the interval reachable by backward portions of arcs, because a given point is reachable by exactly one arc leaving $p_i$ with tangent $t_i$. These regions (if non-empty) may join at the point where an arc is tangent to the gate, but if this arc is invalid (because it intersects the boundary) the regions will be separated. See Figure 10.

### *3.5 Computing Valid Biarcs*

We now look at a fixed pair $p_ip_j$ and test for a valid biarc between $p_i$ and $p_j$. The tangent directions $t_i$ and $t_j$ define a joint circle $J$. For the rest of the paper we will deal with the situation that the first arc starting at $p_i$ is outside the joint circle $J$ and the second arc is inside. The other case is symmetic.

Each gate $g_{i+1}, \ldots, g_{j-1}$ may or may not fulfill the following conditions:

Condition (Out)
The visibility region $W_i$ from $p_i$ intersects the gate OUTSIDE the joint circle.

Condition (In)
The visibility region $\tilde{W}_j$ from $p_j$ intersects the gate INSIDE the joint circle.

We can test the conditions (In) and (Out) in constant time for every gate.

In the following, we will refer to the region bounded by two successive gates $g_{k-1}$ and $g_k$ and the boundary of $R$, as the *cell* between these gates, or simply the *cell* $g_{k-1}, g_k$.

**Lemma 3.4** *(a) If the joint circle contains a joint point for a valid biarc in the cell between $g_{k-1}$ and $g_k$ then $g_{k-1}$ satisfies* (Out)*, and $g_k$ satisfies* (In)*.*
*(b) If $g_k$ satisfies* (In) *then so does $g_{k+1}, \ldots, g_{j-1}$. If $g_k$ satisfies* (Out) *then so does $g_{k-1}, \ldots, g_{i+1}$.*

**PROOF.** We prove only the statements regarding (Out). The arcs starting at $p_i$ in direction $t_i$ start outside the joint circle $J$. If such an arc enters the joint circle, it remains inside until it returns to $p_i$ (see Figure 11). Thus, if

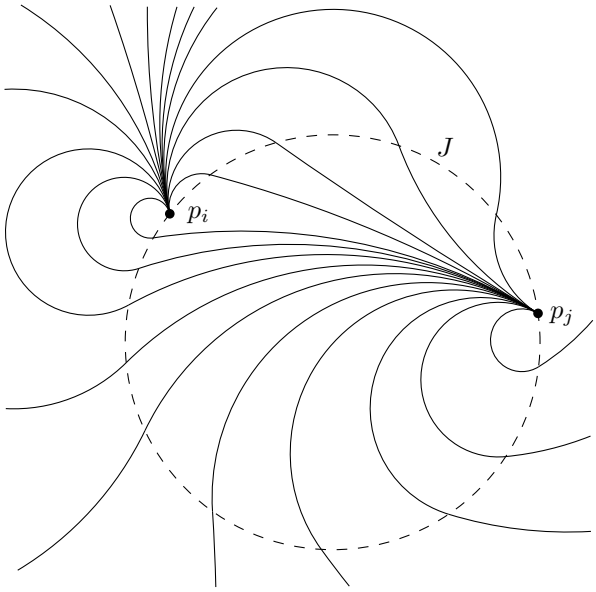


Fig. 11. A family of biarcs from $p_i$ to $p_j$ and its joint circle $J$

an arc has reached $g_k$ outside $J$, the initial part must have passed through $g_{i+1}, \ldots, g_{k-1}$ outside $J$. This establishes part (b) of the lemma. The same argument works for an arc that reaches $J$ in the cell between $g_{k-1}$ and $g_k$ (part (a) of the lemma). □

It follows that the sequence $g_{i+1}, \ldots, g_{j-1}$ can be partitioned into three consecutive parts:

(a) an initial part $g_{i+1}, \ldots, g_{l-1}$ (possibly empty) satisfying (Out) but not (In);
(b) a middle part $g_l, \ldots, g_r$, which is either
  (b1) a nonempty sequence satisfying neither (In) nor (Out);
  (b2) a possibly empty sequence satisfying both (In) and (Out);
(c) a final part $g_{r+1}, \ldots, g_{j-1}$ (possibly empty) satisfying (In) but not (Out).

Since the conditions (In) and (Out) can be tested in constant time, the positions $l$ and $r$ can be identified by binary search in $O(\log n)$ time. From Lemma 3.4a it is clear that in case (b1), there can be no valid biarc, and in case (b2), the joint must be in the cells between $g_{l-1}$ and $g_{r+1}$.

Let us now concentrate on case (b2): We treat the cells $g_{l-1}, g_l$ and $g_r, g_{r+1}$ separately, and test whether some point of $J$ is reachable from $p_i$ and $p_j$, in constant time. (These two cells are the same if the middle part is empty, i.e. $l - 1 = r$.)

It may happen that $g_l$ or $g_r$ intersect $J$ twice, and both pieces outside $J$ are reachable in $W_i$. In this case, it is certain a valid biarc exists, and we need not

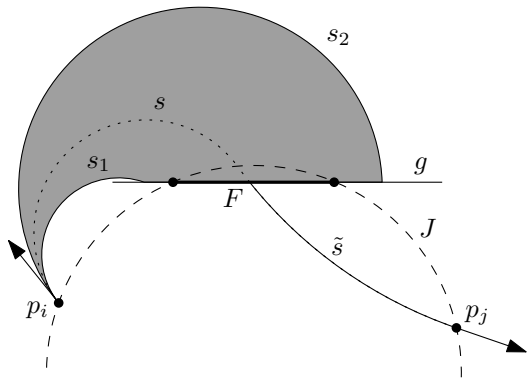


Fig. 12. A joint circle $J$ intersecting a gate $g$ twice. The shaded region is contained in $R$.

proceed.

**Lemma 3.5** *Let $g$ be a gate that satisfies* (In) *and* (Out) *and intersects $J$ twice, and suppose that both pieces outside $J$ are reachable in $W_i$. Then there is a valid biarc between $p_i$ and $p_j$.*

**PROOF.** Let $s_1, s_2$ be two arcs in $W_i$, such that $s_1$ reaches one side of the gate $g$ outside of $J$ and $s_2$ the other. $F$ denotes the segment on $g$ between the two intersections of $J$ (see Figure 12). If $s_1$ or $s_2$ reaches $g$ as part of the backward visibility segment on $g$, then it must reach the same outer piece of $g$ as part of the forward visibility segment. We can thus assume that $s_1$ and $s_2$ extend from $p_i$ until they hit $g$ for the first time. The region bounded by $s_1, s_2$ on segment $g$ is contained within $R$, and therefore the part of $J$ between the intersections with $g$ is completely in $W_i$. Let us sweep the circular arc $s$ from $s_1$ to $s_2$. Each of these arcs is a valid arc and it can be extended to a biarc ending in $p_j$ (not necessarily valid). These biarcs sweep over the segment $F$. By condition (In), we know that at least one of the complementary arcs $\tilde{s}$ is a valid arc, at least to the point where it hits $F$. Since the region between $F$ and $J$ lies within $R$, the whole biarc is in $R$.

We show that this biarc is also valid biarc, since it does not intersect $g_i$ and $g_j$ except at $p_i$ and $p_j$: By construction, the first arc $s$ (up to the joint on $J$) does not intersect $g_i$ twice, since its endpoint is in $W_i$, and it does not intersect $g_j$ at all, since it terminates before crossing $g$. Similarly, the second arc $\tilde{s}$ does not intersect $g_j$ twice, its endpoint being in separated from $g_j$ by $g$. It could possibly intersect $g_i$ only in the circular segment between $F$ and $J$, but since this region is part of $W_i$, this is impossible. □

So, the final case that we have to deal with is the following. We have a sequence of gates $g_l, g_{l+1}, \ldots, g_r$, that satisfy (In) and (Out). We therefore know that $g_l$ and $g_r$ (as well as all intermediate gates) intersect $J$ in points $Q_l$ and $Q_r$

(see Figure 13). It may happen that $g_l$ or $g_r$ intersects $J$ twice, but then only one of the outer parts is intersected by $W_i$. (Otherwise we are done, by the previous lemma.) We denote by $g_l^{\text{out}}$ and $g_r^{\text{out}}$ that outer segment of $g_l$ and $g_r$ that is intersected by $W_i$. The intersection $Q_l$ and $Q_r$ is chosen (in case there are two intersections) as the one that is incident to $g_l^{\text{out}}$ and $g_r^{\text{out}}$, respectively.

We know that there is a valid arc from $p_i$ to $g_r^{\text{out}}$. If the arc reaches $g_r^{\text{out}}$ as part of the backward visiblity, it must have passed through $g_r^{\text{out}}$ as part of the forward visibility region. Thus we denote the first intersection of the arc with $g_r^{\text{out}}$ by $S_r^{\text{out}}$. Similary, there is an arc from $p_j$ that reaches $g_l$ inside $J$ for the first time in some point $S_l^{\text{in}}$, after passing through $g_r$ in the point $S_r^{\text{in}}$ inside $J$.

Now, the two segments $S_l^{\text{in}}S_l^{\text{out}}$ and $S_r^{\text{in}}S_r^{\text{out}}$ and the two circular arcs $S_l^{\text{out}}S_r^{\text{out}}$ and $S_l^{\text{in}}S_r^{\text{in}}$ are contained within $R$. It follows that the four-sided region $G$ enclosed by these curves (shaded in Figure 13) contains no part of the boundary of $R$, and in particular, the arc $Q_lQ_r$ of the joint circle that lies in this region is not intersected by the boundary of $R$.

**Lemma 3.6** *In the situation described above, a joint in the region $R$ between $g_l$ and $g_r$ can only lie on the arc $Q_lQ_r$.*

This lemma seems obvious at first sight, but it is conceivable that this region contains parts of $J$ besides the arc $Q_lQ_r$, as in the example of Figure 13.

**PROOF.** Let $\hat{R}$ denote the region $R$ between gates $g_l$ and $g_r$. We denote by $B_l$ and $B_r$ the endpoints of $g_l^{\text{out}}$ and $g_r^{\text{out}}$, and by $\bar{B}_l$ and $\bar{B}_r$ the opposite endpoints of $g_l$ and $g_r$. Since the region $G$ lies inside $\hat{R}$, it follows that the boundary of $\hat{R}$ must connect $B_l$ with $B_r$ and $\bar{B}_l$ with $\bar{B}_r$. (The opposite connection, $B_l$ with $\bar{B}_r$ and $\bar{B}_l$ with $B_r$, would lead to a crossing.)

A valid arc starting from $p_i$ enters $\hat{R}$ through $g_l^{\text{out}}$. This arc is then in the region $\hat{R}^{\text{out}}$ that is bounded by $g_l^{\text{out}}$, $g_r^{\text{out}}$, the arc $Q_lQ_r$, and the boundary of $\hat{R}$ between $B_l$ and $B_r$. The arc may leave this region through $g_r^{\text{out}}$, but then it has to reenter through $g_r^{\text{out}}$ in order to become a valid arc ending in $\hat{R}$. If the arc hits $Q_lQ_r$ it terminates there. The arc must therefore meet $J$ in the region $\hat{R}^{\text{out}}$.

We can apply a similar argument for the backward arc from $p_j$. This arc is caught in the complementary region $\hat{R}^{\text{in}}$ that is bounded by the arc $Q_lQ_r$, the boundary of $\hat{R}$ between $\bar{B}_l$ and $\bar{B}_r$, and part of the segments $g_l$ and $g_r$. Since the regions $\hat{R}^{\text{out}}$ and $\hat{R}^{\text{in}}$ intersect only in the arc $Q_lQ_r$, the joint can only lie on this arc. □

Now we can easily determine the points on the arc $Q_lQ_r$ that are joints of

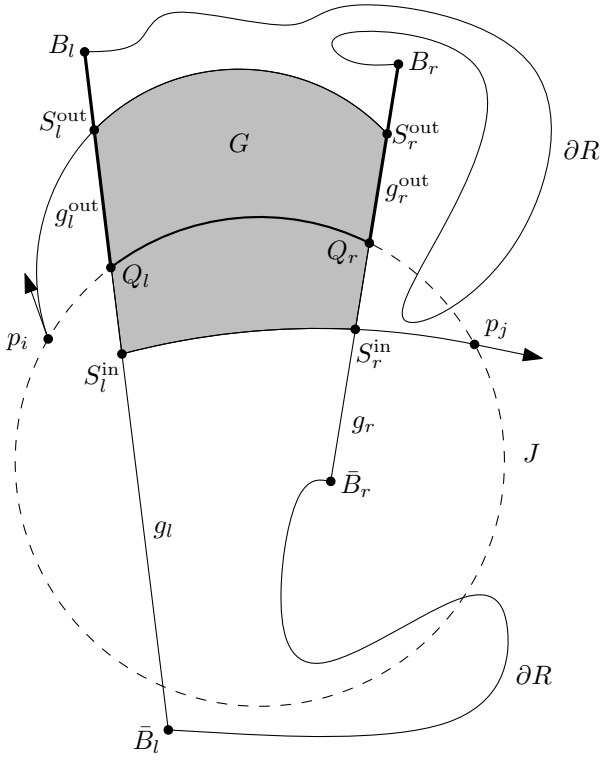


Fig. 13. The region $\hat{R}$ between $g_l$ and $g_r$

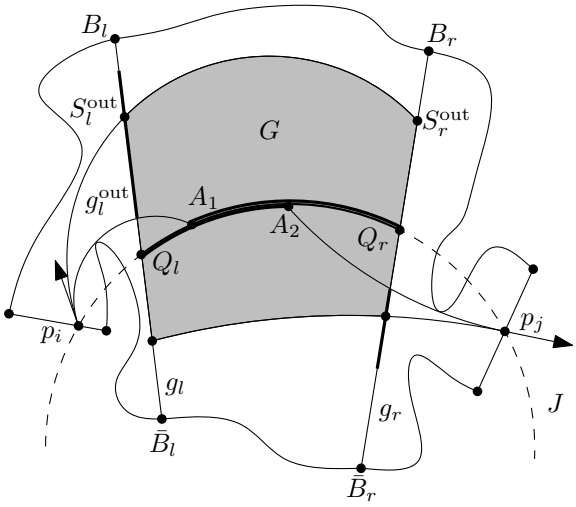


Fig. 14. Determining the valid joint points

valid biarcs.

We have established that the region $G$ does not contain any obstacles. Now consider the point on $g_l^{\text{out}}$ closest to $Q_l$ that lies in $W_i$, and extend the arc from $p_i$ through this point until it hits $J$ in some point $A_1$ (see Figure 14). It may happen that $A_1 = Q_l$ if this point is in $W_i$.

If $A_1$ lies beyond $Q_r$, we conclude that no arc can reach $Q_lQ_r$, because such

an arc would have to intersect $g_l^{\text{out}}$ closer to $Q_l$. Otherwise, by stretching the arc and sweeping out till the arc $p_i S_l^{\text{out}} S_r^{\text{out}}$, we see that the complete interval between $A_1$ and $Q_r$ is reachable from $p_i$ by an arc that stays inside $R$. (It also follows that $A_1$ cannot lie before $Q_l$: by the above argument, $Q_l$ would then be in $W_i$, and $A_1 = Q_l$ would have been chosen instead.)

Similarly, we can look for the point on $g_r$ that lies in $\tilde{W}_j$, inside $J$, and is closest to $Q_r$, and we extend the arc from $p_j$ to a point $A_2$ on $J$. We conclude that the whole sub-arc $Q_l A_2$ is reachable from $p_j$, or that no point on the arc $Q_l Q_r$ is reachable.

By intersecting the arcs $A_1 Q_r$ and $Q_l A_2$, we eventually find the arc $A_1 A_2$ of possible joints, or we find that no joints are possible. The joints on the arc $A_1 A_2$ correspond to biarcs that lie in $R$. To get *valid* biarcs, we have to ensure that they do not intersect $g_i$ and $g_j$ other than in their endpoints. It is straightforward to reduce the interval $A_1 A_2$ in order to exclude the biarcs violating this condition, in constant time.

We summarize what we have achieved.

**Lemma 3.7** *After the intersections of all visibility regions $W_i$ and $\tilde{W}_i$ with all gates $g_k$ have been computed (Lemma 3.3), the existence of a valid biarc between two given endpoints $p_i$ and $p_j$ can be tested in $O(\log n)$ time.*

**PROOF.** We do binary search on the gates between $g_i$ and $g_j$ to find the locations of $g_l$ and $g_r$. This is the part of the procedure that takes $O(\log n)$ time. If $r < l - 1$ there are no valid biarcs between the gates. Otherwise we test the cell between $g_{l-1}$ and $g_l$ and the cell between $g_r$ and $g_{r+1}$ to see if any part of $J$ within these cells is a joint for a valid biarc. (The two cells could be the same cell.) Because the cells have constant complexity these tests can be done in constant time. We check if $g_l$ or $g_r$ intersects $J$ twice and $W_i$ intersects the gate on both sides of $J$. If so there is a valid biarc. If not, we compute $A_1$ and $A_2$ as described above and see if the arc $A_1 A_2$ is non-empty. If not, there is no valid biarc. If so, we reduce it if necessary to eliminate biarcs that intersect $g_i$ or $g_j$ in points other than their endpoints. If any points remain in the interval we report that a valid biarc exists. All work after finding $g_l$ and $g_r$ requires $O(1)$ time. □

With the help of this test, we can now define the directed graph of reachable arcs, and the shortest path will give us the approximation with the fewest biarcs:

**Theorem 3.8** *Given an open polygonal curve $P = (p_1, \ldots, p_n)$, a polygonal tolerance boundary of size $O(n)$, and a gate for each $p_i$, we can approximate*

*$P$ by a minimum number of valid biarcs in $O(n^2 \log n)$ time and $O(n^2)$ space.*

**PROOF.** For each point $p_i$, determine which part of every other gate is reachable by computing $W_i$ and $\tilde{W}_i$ in $O(n^2)$ time and space. For each point pair $p_i, p_j$ we check whether a valid biarc exists, by Lemma 3.7. This requires $O(\log n)$ time, for a total run time of $O(n^2 \log n)$. We use these tests to set up a directed acyclic graph and compute the shortest path, in $O(n^2)$ time. □

As for arcs, there are instances where this bound is overly pessimistic. The visibility regions $W_i$ and $\tilde{W}_j$ will often not extend beyond a few gates, and the calculation can be shortcut.

## 4 The Tolerance Boundary

The "approximation error" $\varepsilon$ for biarcs enters our problem only through the tolerance region $R$. The definition of a useful tolerance boundary for a given curve is a modeling question that depends very much on the application. For some applications, like cutting, it makes sense to use *asymmetric*, one-sided tolerance boundaries. As long the boundary meets the requirements specified in the beginning of this paper, our algorithm can deal with it. Note that the width of the tolerance boundary may change within $R$. Therefore depending on the tolerance boundary our algorithm can answer the classical question for $\varepsilon$ approximation (our solution has absolute guarantees for the minimum number of biarcs that are at most epsilon away from the original curve), as well as for approximations with changing precision requirements. More sensitive parts of the polygonal curve can be approximated with smaller $\varepsilon$ than less important ones. Allowing variations of the width of the approximation boundary has no input on the theoretical complexity of the problem. This ability to vary the width of the approximation boundary makes our algorithm useful for smoothing paths in robotics motion planning. The corridor used for the robotics is usually defined by the obstacles which have to be avoided. This leads naturally to corridors with non-constant width.

**Acknowledgments.** We would like to thank Martin Held for helpful discussions.

## References


[1] K. Bolton. Biarc curves. *Computer-Aided Design*, 7:89–92, 1975.

[2] W. S. Chan and F. Chin. Approximation of polygonal curves with minimum numbers of line segments or minimum error. *International Journal of Computational Geometry & Applications*, 6:59–77, 1996.

[3] F. Chazal and G. Vegter. Computation of the medial axis of smooth curves with topological guarantees. *ACS Technical Report*, ACS-TR-122102-01, 2006.

[4] R. S. Drysdale, G. Rote and A. Sturm. Approximation of an open polygonal curve with a minimum number of circular arcs. In: *Proc. 22nd European Workshop on Computational Geometry (EWCG)*, Delphi, Greece, 2006, pp. 25–28.

[5] J. Eibl. Approximation of Planar Curves Within an Asymmetric Tolerance Band. *M. Sc. thesis, Universität Salzburg, Institut für Computerwissenschaften*, 2002.

[6] D. Eu and G. Toussaint. On approximating polygonal curves in two and three dimensions. *Computer Vision, Graphics and Image Processing*, 56:231–246, 1994.

[7] L. Guibas, J. Hershberger, J. Mitchell and J. Snoeyink. Approximating polygons and subdivisions with minimum-link paths. *International Journal of Computational Geometry & Applications*, 3:383–415, 1993.

[8] M. Held and J. Eibl. Biarc approximation of polygons within asymmetric tolerance bands. *Computer-Aided Design*, 37:357–371, 2005.

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

[10] H. Imai and M. Iri. Polygonal approximation of a curve — formulations and algorithms. In: *Computational Morphology*, G. T. Toussaint, Ed., Elsevier, pp. 71–86, 1988.

[11] D. S. Meek and D. J. Walton. Approximation of discrete data by $G^1$ arc splines. *Computer-Aided Design*, 24:301–306, 1992.

[12] D. S. Meek and D. J. Walton. Approximating quadratic NURBS curves by arc splines. *Computer-Aided Design*, 25:371–376, 1993.

[13] D. S. Meek and D. J. Walton. Approximating smooth planar curves by arc splines. *Journal of Computational and Applied Mathematics*, 59:221–231, 1995.

[14] A. Melkman and J. O'Rourke. On polygonal chain approximation. In: *Computational Morphology*, G. T. Toussaint, Ed., Elsevier, pp. 87–95, 1988.

[15] L. Piegl. Curve fitting for rough cutting. *Computer-Aided Design*, 18:79–82, 1986.

[16] J. Schönherr. Smooth biarc curves. *Computer-Aided Design*, 25:365–370, 1993.

[17] B.-Q. Su and D.-Y. Liu. *Computational Geometry — Curve and Surface Modeling*. Academic Press, 1989.

[18] G. T. Toussaint. On the complexity of approximating polygonal curves in the plane. *Proceedings IASTED, International Symposium on Robotics and Automation, Lugano, Switzerland*, 1985.

[19] K. R. Varadarajan. Approximating monotone polygonal curves using the uniform metric. In: *Proc. 12th Ann. Symp. on Computational Geometry (SCG)*, 1996, pp. 311–318.

[20] M. Yeung and D. J. Walton. Curve fitting with arc splines for NC toolpath generation. *Computer-Aided Design*, 26:845–849, 1994.

[21] J.-H. Yong, S.-M. Hu and J.-G. Sun. A note on approximation of discrete data by $G^1$ arc splines. *Computer-Aided Design*, 31:911–915, 1999.