

RESEARCH ARTICLE

Connect the dot: Computing feed-links for network extension*

Boris Aronov¹, Kevin Buchin², Maike Buchin², Bart Jansen³,
Tom de Jong⁴, Marc van Kreveld³, Maarten Löffler⁵, Jun Luo⁶,
Rodrigo I. Silveira⁷, and Bettina Speckmann²

¹Department of Computer Science and Engineering, Polytechnic Institute of NYU, USA

²Dept. of Mathematics and Computer Science, Eindhoven University of Technology, Netherlands

³Department of Information and Computing Sciences, Utrecht University, Netherlands

⁴Department of Geography, Utrecht University, Netherlands

⁵Department of Computer Science, University of California at Irvine, US

⁶Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, China

⁷Departament de Matemàtica Aplicada II, Universitat Politècnica de Catalunya, Spain

Received: February 14, 2011; accepted: May 6, 2011

Abstract: Road network analysis can require distance from points that are not on the network themselves. We study the algorithmic problem of connecting a point inside a face (region) of the road network to its boundary while minimizing the detour factor of that point to any point on the boundary of the face. We show that the optimal single connection (feed-link) can be computed in $O(\lambda_7(n) \log n)$ time, where n is the number of vertices that bounds the face and $\lambda_7(n)$ is the slightly superlinear maximum length of a Davenport-Schinzel sequence of order 7 on n symbols. We also present approximation results for placing more feed-links, deal with the case that there are obstacles in the face of the road network that contains the point to be connected, and present various related results.

Keywords: network analysis, geometric algorithms, feed-links, shortest path, dilation

1 Introduction

In geographical context, network analysis is a type of geographical analysis on real world networks, such as road, subway, or river networks. Many facility location problems involve

*This paper is a merged and expanded version of two conference papers that appeared at ACM SIGSPATIAL 2008 [3] and WADS 2009 [4].

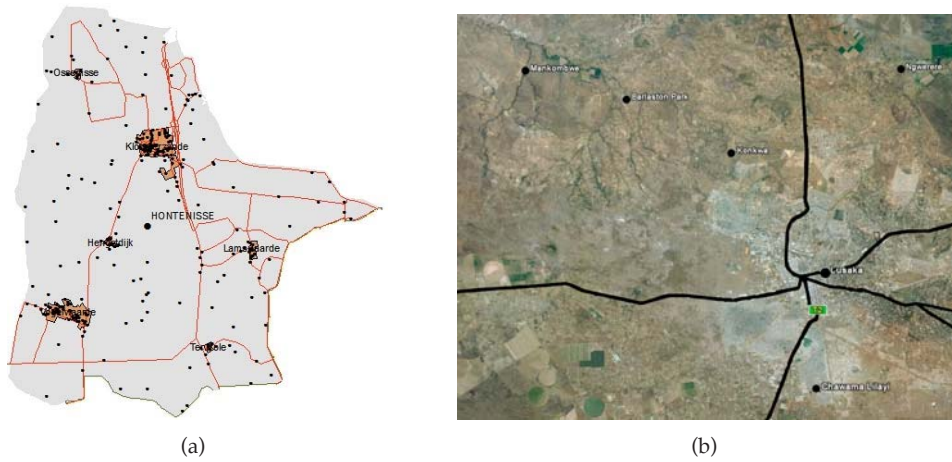


Figure 1: (a) Road network of the Dutch municipality of Hontenisse, showing the centers of the postal code areas (black dots), most of them missed by the roads. (b) Villages around Lusaka, Zambia, appear disconnected from the road network of Google Earth because smaller roads are missing in the data.

network analysis. For example, when a location for a new hospital needs to be chosen, a feasibility study typically includes values that state how many people would have their travel time to the nearest hospital decreased to below 30 minutes due to the new hospital location. In geographic accessibility studies, one may analyze how many households are reachable within 45 minutes from a fire station. In this case, the households are typically aggregated by municipality or postal-code region, and the centroid of this region is taken as the representative point. This representative point usually does not lie on the road network, as seen in the example in Figure 1(a). It might even be far removed from it, since nationwide accessibility studies rarely use detailed network data for their analysis. A similar situation occurs when the quality of the network data is not very high. In developing countries, data sets are often incomplete due to omissions in the digitization process, or due to lack of regular updates, as seen in Figure 1(b). Even when road network data exists, it may not be available in full detail for many different reasons. National or provincial authorities are often responsible only for roads of a certain level and do not record local or smaller roads in their databases.

When performing network analysis, it is necessary for all points of importance to be connected to the network. A number of different solutions to this problem are used in practice. The simplest approach physically snaps the locations of these points to the network. This may produce very unrealistic results if two nearby locations, inside the same face of the network, are snapped to opposite sides of that face. Moreover, this approach modifies the actual positions of the locations, which we assumed to be correct. A different strategy places links from all locations inside a face to a central location, a so-called *feed-node*, which is connected directly to the network. The feed-node can be the centroid of the face or some other relevant point. Another approach connects each individual location to a network node or segment using a so-called *feed-link*. This approach is taken by Dahlgren [6], Dahlgren and Harrie [7, 8], and Ness and Brogaard [21], who choose the

nearest network location to connect to. Alternatively, important locations can be connected to the road network using a Delaunay triangulation of the important locations and nodes on the road network. De Jong and Tillema [12] take this approach. They discard Delaunay edges that cross obstacles and use the remaining Delaunay edges as part of the road network.

A feed-link is an artificial connection between a location and the known network that is “reasonable.” When a connection is “reasonable” is subject to interpretation. One possibility is to say that it means it is conceivable that such a connection exists in the real world. More generally, we may also consider a connection to be “reasonable” if it results in a similar network distance as could be assumed in the real (unknown) network, even if the actual connection at that particular place is not probable, because when doing network analysis, the distances are what is important. Previous work that considers feed-links to repair networks with disconnected locations makes no attempt to verify or quantify how “reasonable” the resulting connections are [6, 8, 12, 21].

In this paper, we propose to use the concept of *dilation* to quantify the quality of feed-links. This concept (defined below) informally captures the amount of detour between any two points on the network. People in general do not like detours, so a connection that causes as little detour as possible is more likely to be “real.” Previous studies in developed countries have shown that the average detour of road networks usually varies between 1.13 and 1.45 [22, 23]. Networks in developing countries typically have a higher coefficient. Also in areas with large natural obstacles, like lakes and mountain ranges, the dilation is usually larger.

Given an embedded planar graph, we define the *detour*, also known as *crow flight conversion coefficient*, of two points p and q on the graph to be the ratio between their distance on the graph and their Euclidean or crow flight distance. We consider both the vertices and arbitrary points on edges. The *geometric dilation* of the graph is the worst (maximum) detour between any pair of points on the graph. For example, the square grid, illustrated in Figure 2(a), has a dilation of 2, because for any two points on the grid lines, a shortest path is at most twice as long as the Euclidean distance. The dilation of 2 is realized by the midpoints of opposite sides of a square cell. An equilateral triangle also has a dilation of 2, while a circle has a dilation of $\pi/2 \approx 1.57$, see Figure 2. It is known that any network containing a closed loop cannot have geometric dilation better than $\pi/2$, and any set of points can be extended to a network that has a geometric dilation no more than 1.68 [13]. A related concept in computational geometry is that of a *t-spanner*, which is a graph defined on a set of points such that the detour between any pair of those points is at most t , see [1, 16, 19, 20]. For spanners it is more common to consider *graph-based dilation*, where we consider dilation only at the vertices (input points) of the spanner, not at points on the edges.

In road networks, disconnected points occur in faces of the network. It seems reasonable to choose feed-links from a disconnected point to some point on the boundary of the face it occurs in. This motivates the problem we discuss in the remainder of this paper: given a boundary P of a simple polygon, possibly with obstacles inside it, and a point p inside P , how can we connect it to P with one or more feed-links while ensuring a small detour between p and any point on P ? With a slight abuse of terminology, we will also refer to this worst detour from p as *dilation* in the remainder.

The rest of this paper is organized as follows. In Section 2 we present a precise formulation of the problem and discuss several modeling choices. We also show that, if a set

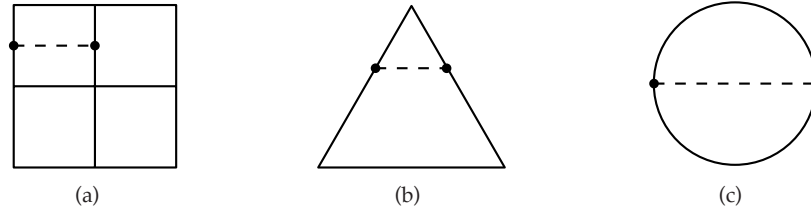


Figure 2: Dilation in (a) a square, (b) an equilateral triangle, and (c) a circle. The dashed lines indicate pairs of points achieving the dilation.

of feed-links between p and P is given, then the dilation can be computed efficiently. In Section 3 we present an algorithm that computes a single feed-link from a given point in a given simple polygon optimally. Then we extend the algorithm to deal with obstacles. In Section 4 we study the problem of placing multiple feed-links, and discuss simple polygons, convex polygons, and realistic polygons. In Section 5 we present simple heuristics for placing feed-links in practice, and evaluate them experimentally. In Section 6 we conclude by summarizing our results and giving directions for future research.

2 Preliminaries

In this section, we formalize the concepts of feed-links and dilation. First, we define several concepts and notation needed for the problem formulation. We also motivate our choice of the precise definition of dilation. Second, we show how to compute the dilation if feed-links are given.

2.1 Notation and problem statement

We assume the (road) network is given as an embedded planar graph. Hence, a location p that does not lie on the network, lies inside some face of this graph. If p lies inside a bounded face and the network is biconnected, then the face in which p lies can be represented by a simple polygon whose boundary we denote by P . From now on, P and p will always refer to a polygon and a special point inside it that we wish to connect. A feed-link is a straight-line segment from p to some point q on P . We are interested in achieving a small detour from any point on P to p by placing one or more feed-links suitably.

For two points a and b on P , $P[a, b]$ denotes the portion of P from a clockwise to b ; its length is denoted by $\mu(a, b)$. Furthermore, $\mu(P)$ denotes the length (perimeter) of P . The Euclidean distance between two points a and b is denoted by $|ab|$. The shortest path in the network (including feed-links) between two points a and b is denoted by $\text{sp}(a, b)$ and its length is denoted by $|\text{sp}(a, b)|$. Obviously, from any point r on P , the shortest path $\text{sp}(r, p)$ uses exactly one feed-link. The detour from point r to point p is $|\text{sp}(r, p)|/|pr|$.

Now assume that a single feed-link \overline{pq} exists between p and a point q on P . The detour of r (to p) is denoted $\delta_q(r)$ and it is equal to $|\text{sp}(r, p)|/|pr| = (|\text{sp}(r, q)| + |pq|)/|pr| = (\min(\mu(r, q), \mu(q, r)) + |pq|)/|pr|$. For a subset $R \subset P$ of locations on P , we denote by $\delta_q(R) = \max_{r \in R} \delta_q(r)$ the worst detour any point in R has to p , for this particular feed-link \overline{pq} . Note that $\delta_q(q) = 1$. A single feed-link is placed optimally if it minimizes $\delta_q(P)$ over all

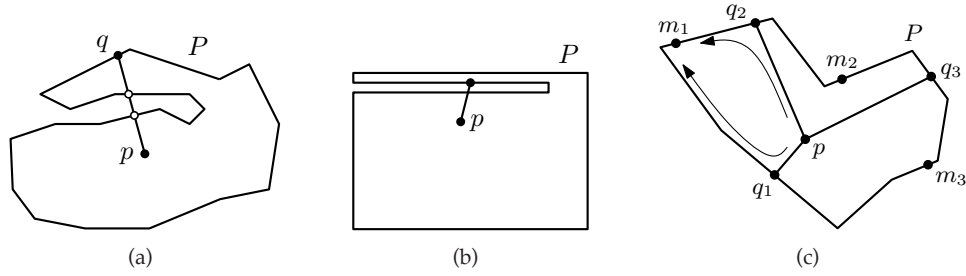


Figure 3: (a) A feed-link that intersects P gives no access to the feed-link other than q . (b) A minimum dilation feed-link may intersect P in the interior of the feed-link. (c) At the points m_1, m_2 , and m_3 , the used feed-link changes.

choices of q . Here we consider all possible points on P , not just the vertices: a feed-link can connect to any point and the dilation is measured from any point.

As stated above, we assume that a feed-link is a straight-line connection between p and exactly one point q on P . Since P can be any simple polygon, a feed-link \overline{pq} can intersect P in more points, see Figure 3(a), but we assume that it is not possible to “hop on” the feed-link at any such point other than q (the white points in the figure provide no access to the feed-link). Figure 3(b) shows that the feed-link yielding minimum dilation may still intersect P in a point other than q . We can imagine that the feed-link is actually a tunnel going under the existing surface. While this is not a very realistic situation, we motivate the use of the tunnel view by noting that it is the resulting distances of the network we are interested in, and not the actual layout of the resulting network. Furthermore, the alternatives to this view all have some limitations:

- One could allow multiple access points on a single feed-link. The drawback is that this is in a way unfair, because with a single feed-link many connections may be formed. This could result in an unrealistic bias towards creating feed-links that intersect the network very often.
- One could choose to only allow feed-links to points visible from p . The drawback is that a solution as in Figure 3(b) would not be possible anymore, limiting the freedom of where to place feed-links.
- One could use geodesic shortest paths inside P as feed-links, essentially treating the exterior of P as an obstacle. It is then natural to measure the dilation of any point on P with respect to its geodesic distance to p as well, instead of using its Euclidean distance. The drawback of this view is that there is often no natural reason for disallowing a connection outside P .

Although we will adopt the tunnel view in this paper, several of our results can also be extended to other models. In particular, the second alternative implies restricting the possible locations to connect a feed-link, whereas the third alternative is a special case of the situation with obstacles, discussed below. Our algorithms can generally not be adapted to the first alternative.

Let us consider the case where there are obstacles in P , which could model impassable mountains or lakes. We represent the obstacles by simple polygons and denote them by L_1, \dots, L_h . The obstacles are not allowed to intersect P or contain p . A feed-link from any

point q on P is now a shortest path from q to p that avoids the obstacles L_1, \dots, L_h . The detour of a point r on P is now defined as the ratio between the shortest network path and the shortest obstacle-avoiding path from r to p . Observe that for a point q that has a feed-link, the detour $\delta_q(q) = 1$, just like in the case without obstacles. We also note that the third alternative to the tunnel view can be modeled by adding the exterior of P as an obstacle.

2.2 Computing the dilation

Before we study algorithms for placing feed-links, we show in this section that we can compute the dilation for a given set of feed-links efficiently. First, we study the situation without obstacles, and then we extend the approach to handle obstacles as well.

Assume that the points q_1, \dots, q_k where the feed-links attach to P are sorted along P . For any two consecutive points q_i and q_{i+1} , find the point m_i on P where the network distance to p via feed-link $\overline{pq_i}$ is equal to the network distance via feed-link $\overline{pq_{i+1}}$ (see Figure 3(c)). Then along P , we have points $q_1, m_1, q_2, m_2, \dots, q_k, m_k$. All points between m_j and m_{j+1} will have their best network connection to p via q_{j+1} .

For any point on an edge of P between m_j and m_{j+1} , the network distance changes linearly in the position of that point on the edge, and the Euclidean distance changes hyperbolically. Therefore, an analytic computation can determine the location on the edge where the dilation is realized: if we parameterize the edge by $t \in [0, 1]$, then the network distance is a linear function $at + b$ where $a > 0$ and $b \geq 0$ are reals depending only on P , p , and q_{j+1} . The Euclidean distance has the form $\sqrt{At^2 + Bt + C}$ where A , B , and C are constants depending only on the coordinates of p and the endpoints of the edge. By setting the derivative of the quotient to zero, we get as parameter values of a possible maximum $t = (-bB + 2aC)/(2Ab - aB)$, which we insert into the quotient to determine the detour at the corresponding point on the edge, and check if it is larger than any detour found so far.

The computation of the maximum is done for all edges between m_j and m_{j+1} , and similarly, for all pairs of consecutive midpoints of this type.

If P has n edges, then splitting edges at the points q_1, \dots, q_k and m_1, \dots, m_k gives rise to at most $n + 2k$ edges on which we maximize the detour, taking constant time for each. Therefore, we can compute the dilation of the polygon and its feed-links in $O(n + k)$ time. If we need to compute the sorted order of q_1, \dots, q_k on the boundary, we must add an $O(k \log k)$ term. However, note that k is typically a small constant, and the dependency on n is the relevant part.

Theorem 2.1. *Given the boundary P of a simple polygon with n vertices and a set of k feed-links, we can compute the dilation in $O(n + k \log k)$ time.*

The situation is more difficult if there are obstacles inside P . Let b be the total number of vertices of the obstacles L_1, \dots, L_h . We can use the algorithm of Hershberger and Suri [18] to find shortest paths amidst obstacles in a scene of complexity n in $O(n \log n)$ time. Since we are interested in shortest paths to p only, we run this algorithm on p and the obstacles L_1, \dots, L_h . The algorithm will compute a subdivision of the whole plane into cells where the first vertex of an obstacle on the shortest path to p (called *anchor* of the path) is fixed. It also computes the geodesic distance from every obstacle vertex to p . The subdivision S has complexity $O(b)$ and can be computed in $O(b \log b)$ time [18].



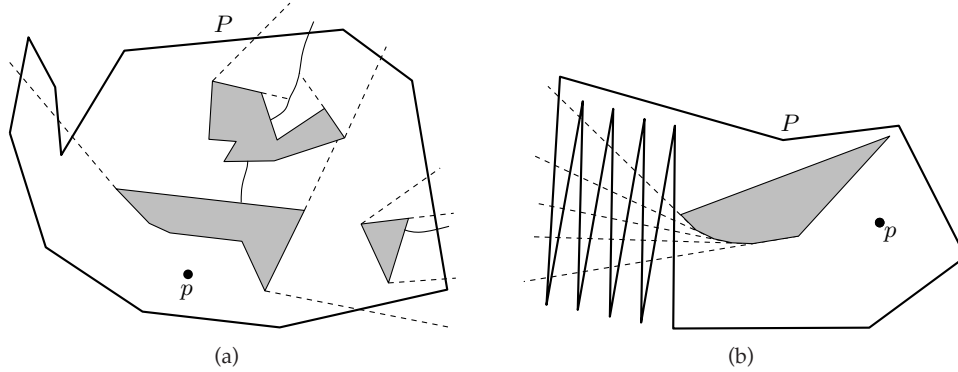


Figure 4: (a) Example of the overlay of subdivision S and the polygon boundary P . The solid (curved) edges divide cells where the topology of the shortest paths change in the sense that when one crosses such edge, the shortest path has to be routed differently around obstacles; the dashed edges divide cells where only the anchor of the shortest path changes. (b) Theoretically, the overlay of S and P can have $\Theta(nb)$ complexity. In practice, this is unlikely.

We overlay the subdivision S with P to partition P 's edges into subedges that have a similar shortest path to p , similar in the sense that the anchor of the shortest path to p is the same (see Figure 4(a)). (Several standard algorithms for computing the overlay can be found in books on computational geometry, e.g. [10].) This allows us to get an analytic expression for the length of the shortest path from any point on P to p . The expression of the length now has the form $\sqrt{At^2 + Bt + C} + D$ for real constants A, B, C , and D , where D is the distance from p to the anchor of the path. We can now find the dilation of the set of feed-links by combining it with the ideas from the case where no obstacles were present. The different analytical expression gives rise to two candidate solutions for the maximum detour, in contrast to the case of no obstacles where there is one candidate.

Theorem 2.2. *Given the boundary P of a simple polygon with n vertices, a set of obstacles with b vertices, and a set of k feed-links, we can compute the dilation in $O(nb \log(nb) + k \log k)$ time.*

Proof. The terms in the time bound are clear from the steps needed in the algorithm. We need $O(b \log b)$ time to compute S . The $O(nb \log(nb))$ term is caused by the overlay of S and P : There may be $O(nb)$ intersection points in the overlay, and therefore the edges of P are partitioned into $O(nb)$ pieces due to the shortest path subdivision. For each piece, we can find the point with maximum detour on that piece in constant time. \square

Note that the algorithm above also finds the closest point on P to the point p in $O(nb)$ time in a polygon with obstacles, which we can use to compute a good feed-link. Without obstacles, this operation can easily be done in $O(n)$ time. We also note that even for one obstacle with b vertices, the overlay of S and P can have $\Theta(nb)$ complexity in theory. However, to obtain this complexity, the polygon and obstacles have to be laid out in very contrived configurations, such as the one depicted in Figure 4(b), and in practice we expect the complexity of the overlay to be much lower.

3 Placing one feed-link . . .

This section discusses the problem of placing one feed-link to p in order to get a low dilation. We present a linear-time, factor-2 approximation algorithm, an algorithm that finds an optimal placement for a single feed-link and runs in $O(\lambda_7(n) \log n)$ time (where $\lambda_7(n)$ is the slightly superlinear maximum length of a Davenport-Schinzel sequence of order 7 on n symbols), and give adaptations to deal with obstacles in the polygon. We also give a linear-time approximation scheme for convex polygons.

3.1 . . . in a simple polygon . . .

An obvious choice for a single feed-link from p to P is to connect p to the closest point on P . We show here that this is indeed a reasonable choice in that it results in at most twice the dilation of the optimal connection.

Lemma 3.1. *If p has one feed-link to the closest point on P , then the resulting dilation obtained by the feed-link is never worse than twice the dilation obtained by an optimally placed feed-link.*

Proof. Suppose that the closest point is q , and that a point r has the worst detour when a feed-link between q and p is chosen. Since q is the only feed-link the detour $\delta_q(r)$ of r is

$$\delta_q(r) = \frac{|\overline{pq}| + |\text{sp}(r, q)|}{|\overline{pr}|}.$$

We need to prove that for any other feed-link q' there is a point r' with detour at least $\delta_{q'}(r') \geq \delta_q(r)/2$. For this we consider several cases depending on which part of the boundary a feed-link connects to. Let m be the point in the middle of the shorter boundary path from r to q (see Figure 5(a)).

Case 1: q' is between q and m . Then the shortest path from q' to r is at most halved. We choose $r' = r$ and have:

$$\delta_{q'}(r) = \frac{|\overline{pq'}| + |\text{sp}(r, q')|}{|\overline{pr}|} \geq \frac{|\overline{pq}| + \frac{1}{2}|\text{sp}(r, q)|}{|\overline{pr}|} \geq \frac{1}{2} \frac{|\overline{pq}| + |\text{sp}(r, q)|}{|\overline{pr}|} = \frac{1}{2} \delta_q(r).$$

In particular, this is true if $q' = m$.

Case 2: q' is between m and r . Then the detour of q is at least as large as for a feed-link to m . We choose $r' = q$ and have:

$$\delta_{q'}(q) \geq \delta_m(q) \geq \delta_m(r) \geq \frac{1}{2} \delta_q(r).$$

Finally, if a feed-link connects to a point on the longer boundary part between q and r , then the same arguments apply. \square

The bound in Lemma 3.1 is tight in the sense that the factor by which the dilation using the closest point is worse than the dilation for the optimal feed-link can be arbitrarily close to 2. This is illustrated in Figure 5(b): q and r are the closest points to p and taking a feed-link to one of them gives a detour of $4x + 7$ at the other. The optimal feed-link is between p and q' and gives a dilation of $2x + 5$ (obtained at q and r). Thus, for $x \rightarrow \infty$, the approximation factor converges to 2.

We have proven the following result.



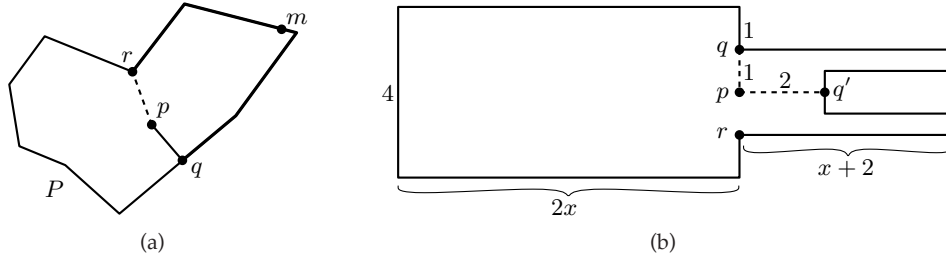


Figure 5: (a) Illustration of the proof of Lemma 3.1: connecting p to its closest point results in a dilation at most twice the optimal one. (b) Example where choosing the closest point to connect the feed-link results in a dilation close to two times worse than when connecting to the optimum feed-link point.

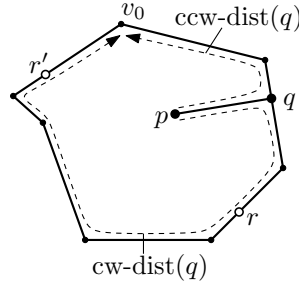


Figure 6: $cw-dist(q)$ and $ccw-dist(q)$; shown is case 1 with order $v_0 q r r'$.

Theorem 3.2. *Given the boundary P of a simple polygon with n vertices, a feed-link that gives a dilation at most twice the optimum can be computed in $O(n)$ time.*

We proceed to show that we can actually place the link optimally and efficiently as well. We will first consider the situation where we only measure the detour at a discrete subset $R \subset P$ of m points on P . After that, we extend the approach to the full continuous case. In both cases, the feed-link may connect to any point on P .

Let v_0, \dots, v_{n-1} be the vertices of P and let p be a point inside P . We seek a point q on P such that the feed-link \overline{pq} minimizes the dilation.

Let r be a point on P and let r' be the point opposite r , that is, the distance along P between r and r' is exactly $\mu(r, r') = \mu(r', r) = \mu(P)/2$. For any given location of q , r has a specific detour. We study the change in detour of r as q moves along P . If $q \in P[r', r]$, then the graph distance between p and r is $|pq| + \mu(q, r)$, otherwise it is $|pq| + \mu(r, q)$.

We fix a point v_0 and define two functions $cw-dist(q)$ and $ccw-dist(q)$ that measure the distance from p to v_0 via the feed-link \overline{pq} and then from q either clockwise or counterclockwise along P , see Figure 6. The detour $\delta_q(r)$ of r can be expressed using either $cw-dist(q)$ or $ccw-dist(q)$, depending on the order in which v_0, q, r , and r' appear along P . In particular, we distinguish four cases that follow from the six possible clockwise orders of v_0, q, r , and r' :

1. If the clockwise boundary order is $v_0 q r r'$ or $v_0 r' q r$, then the detour is $\delta_q(r) = (cw-dist(q) - \mu(r, v_0)) / |pr|$.

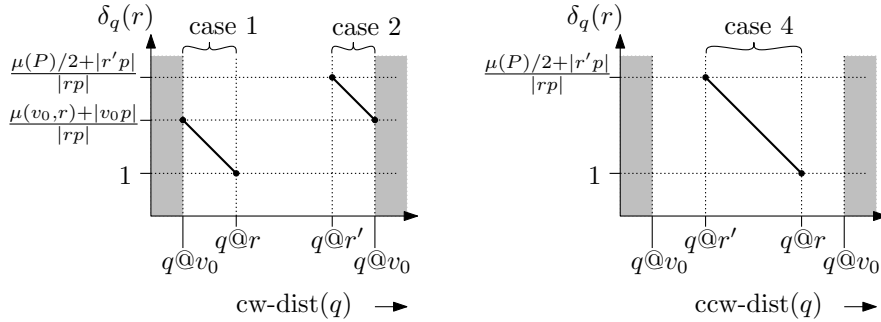


Figure 7: Two graphs showing the detour of a point r as a function of $\text{cw-dist}(q)$ (left) and $\text{ccw-dist}(q)$ (right); $q@r$ indicates “ q is at position r .”

2. If the clockwise boundary order is $v_0 r r' q$, then the detour is $\delta_q(r) = (\text{cw-dist}(q) + \mu(v_0, r)) / |pr|$.
3. If the clockwise boundary order is $v_0 q r' r$, then the detour is $\delta_q(r) = (\text{ccw-dist}(q) + \mu(r, v_0)) / |pr|$.
4. If the clockwise boundary order is $v_0 r q r'$ or $v_0 r' r q$, then the detour is $\delta_q(r) = (\text{ccw-dist}(q) - \mu(v_0, r)) / |pr|$.

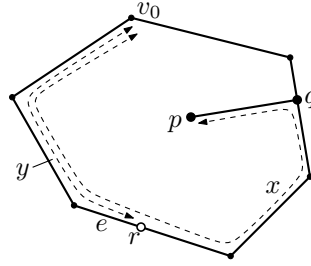
As q moves along P in clockwise direction, starting from v_0 , three of the cases above apply consecutively. Either we have $v_0 q r r' \rightarrow v_0 r q r' \rightarrow v_0 r r' q$, or $v_0 q r' r \rightarrow v_0 r' q r \rightarrow v_0 r' r q$. We parameterize the location of q both by $\text{cw-dist}(q)$ and $\text{ccw-dist}(q)$. This has the useful effect that the detour $\delta_q(r)$ of r is a linear function on the intervals where it is defined (see Figure 7). In particular, for a fixed point r , $\delta_q(r)$ consists of three linear pieces. Note that we cannot combine the two graphs into one, because the parameterizations of the location of q by $\text{cw-dist}(q)$ and $\text{ccw-dist}(q)$ are not linearly related. This follows from the fact that $\text{cw-dist}(q) + \text{ccw-dist}(q) = \mu(P) + 2 \cdot |pq|$.

We now solve the restricted case of minimizing the dilation only for a set R of m given points on P . For each point $r \in R$ we determine the line segments in the two graphs that give the detour of r as a function of $\text{cw-dist}(q)$ and $\text{ccw-dist}(q)$. These line segments can be found in $O(n + m)$ time in total. Next, we compute the upper envelope¹ of the line segments in each of the two graphs. This takes $O(m \log m)$ time using the algorithm of Hershberger [17], and results in two upper envelopes with complexity $O(m \cdot \alpha(m))$, where $\alpha(m)$ is the inverse Ackermann function, an extremely slow growing function that in practice can be considered a constant.

Finally, we scan the two envelopes simultaneously, one from left to right and the other from right to left, taking the maximum of the corresponding positions on the two upper envelopes, and recording the lowest value encountered. This is the optimal position of q .

To implement the scan, we first add the vertices of P to the two envelopes. Since we need to compute the intersection points of the two envelopes we must unify their parameterizations. Consider the locations of q that fall within an interval I which is determined by two envelope edges e_1 and e_2 . Since $\text{cw-dist}(q) = -\text{ccw-dist}(q) + 2 \cdot |pq| + \mu(P)$, the line segment of one envelope restricted to I becomes a hyperbolic arc in the parametrization of the

¹The *upper envelope* $U(x)$ of a set of partial real-valued continuous functions $\{f_i\}$ is the function defined as the pointwise maximum of the functions f_i : $U(x) = \max\{f_i(x) \mid f_i(x) \text{ defined}\}$.

Figure 8: Parameterizing q and r .

other envelope. Hence e_1 and e_2 can intersect at most twice in a unified parametrization, and the scan takes time linear in the sum of the complexities of the two envelopes.

Theorem 3.3. *Given the boundary P of a simple polygon with n vertices, a point p inside P , and a set R of m points on P , we can compute the feed-link (which might connect to any point on P) that minimizes the dilation from p to any point in R in $O(n + m \log m)$ time.*

Next we extend our algorithm to minimize the dilation over all points on P . Let $r_e(q)$ denote the point with the maximum detour on a given edge e of P . For an edge e , $\delta_q(e)$ denotes the maximum detour of any point on e when the feed-link is \overline{pq} .

Instead of considering the graphs of the dilation for a set of fixed points, we consider the graphs for the points $r_e(q)$ for all edges of P . The positions of $r_e(q)$ change with q . The graphs of the dilation do not consist of line segments anymore, but of more complex functions, which, as we will show below, intersect at most six times per pair. As a consequence, we can compute their upper envelope in $O(\lambda_7(n) \log n)$ time [17], where $\lambda_7(n)$ is the maximum length of a Davenport-Schinzel sequence of order 7 on n symbols, which is slightly superlinear [2, 24].

We now argue that the detour of $r_e(q)$ as a function of $\text{cw-dist}(q)$ or $\text{ccw-dist}(q)$ is “well behaved”; that is, any two such functions intersect at most six times. To simplify notation, we denote $\text{cw-dist}(q)$ by x and $\mu(r, v_0)$ by y for any point r on edge e . For the remainder of this argument we assume that case 1 applies, the other cases can be handled analogously. The detour of r is given by

$$\delta_q(r) = \frac{x - y}{|pr|} = \frac{x - y}{\sqrt{ay^2 + by + c}}$$

for constants a, b, c , as long as y is such that r lies on e . Denoting the detour of $r_e(q)$ by $\delta_q(e)$ we further have

$$\delta_q(e) = \max_{r \in e} \delta_q(r) = \max_y \frac{x - y}{\sqrt{ay^2 + by + c}}.$$

To compute the maximum, we compute the derivative with respect to y and set it to zero. This gives $y = \frac{-bx - 2c}{2ax + b}$, which we substitute into the formula for $\delta_q(e)$:

$$\delta_q(e) = \frac{2ax^2 + 2bx + 2c}{\sqrt{x^2(a^2b^2 - 2ab^2) - b^3x + 4ac^2 - 2b^2c + c}}.$$

When we have two such functions in x for different edges e_1 and e_2 , we obtain their intersection points by setting $\delta_q(e_1) = \delta_q(e_2)$. To solve this equation we have to find the

roots of a polynomial of degree six, which implies that two functions have at most six intersection points. Hence, the upper envelope of n of these functions has complexity $O(\lambda_8(n))$, and can be computed in $O(\lambda_7(n) \log n)$ time [17].

In case the maximum is not attained for values of y where r lies on e , the dilation occurs at an endpoint of e . We can simply add the dilation functions of all vertices to the set of functions of which we compute the upper envelope. Similarly, we add the dilation function of the variable point that is exactly opposite from q on P .

Using Hershberger's algorithm [17] to compute envelopes, and scanning the envelopes as before, proves Theorem 3.4.

Theorem 3.4. *Given the boundary P of a simple polygon with n vertices and a point p inside P , we can compute the feed-link that minimizes the dilation from p to any point on P in $O(\lambda_7(n) \log n)$ time.*

Note that our algorithms ignore the degenerate case where p lies on a line supporting an edge e of P . In this case $\text{cw-dist}(q)$ and $\text{ccw-dist}(q)$ are both constant on e . This is in fact easy to handle, as we describe below when discussing dilation in the presence of obstacles.

We can adapt our algorithms to not allow feed-links that intersect the exterior of P , or more generally, to only allow feed-links that connect to a given subset $Q \subset P$. In case we want to disallow intersections, we compute Q by first computing the visibility polygon $V(p)$ of p with respect to P . The vertices of $V(p)$ partition the edges of P into parts that are allowed to contain q and parts that are not. The number of parts is $O(n)$ in total, and they can be computed in $O(n)$ time.

Given a subset $Q \subset P$, we compute the upper envelopes exactly as before. Before we start scanning the two envelopes, we add the vertices of P and also the vertices of Q to the two envelopes. The envelopes now have the property that between two consecutive vertices, a feed-link is allowed everywhere or nowhere. During the scan, we keep the maximum of the dilation functions and record the lowest value that is allowed. The time complexity of our algorithms does not change.

3.2 ... with obstacles

We can also adapt our algorithms to work in the presence of obstacles. As in Section 2.2 where we determined the dilation of a set of feed-links, we use the result of Hershberger and Suri [18] to compute the subdivision S for p and the obstacles L_1, \dots, L_h .

We first show that an obstacle-avoiding feed-link that yields a dilation at most twice the optimum can be determined in $O((n+b) \log(n+b))$ time; recall that b is the total number of vertices of the obstacles. To this end, we observe that Lemma 3.1 still holds for a simple polygon with obstacles if we replace $|pq|$ and $|pr|$ by the geodesic distances between p and q , and p and r , respectively. The proof is exactly the same.

To find the closest point on P to p , we add the exterior of P as an obstacle to the set L and run the algorithm of Hershberger and Suri [18]. The reason for adding P to the obstacles is to avoid the quadratic size overlay; note that since we are only interested in the closest point on P anyway, this is no problem. We stop the algorithm as soon as we find a point on P that is closest to p in the geodesic sense. A feed-link to this point provides the approximation.



Theorem 3.5. *Given the boundary P of a simple polygon with n vertices and a set of obstacles with b vertices in total, a feed-link that gives a dilation at most twice the optimum can be computed in $O((n + b) \log(n + b))$ time.*

We next show that the algorithm that computes an optimal feed-link can be adapted as well. We compute the subdivision S of p and the obstacles L_1, \dots, L_h as before, and overlay it with P . This results in a partition of P into $O(nb)$ parts, and within each part the shortest obstacle-avoiding path to p from a point on that part has the same anchor.

Next we use the algorithm presented earlier in this section for the case without obstacles. When we use $\text{cw-dist}(q)$ and $\text{ccw-dist}(q)$ to represent the location of q , we use the length of the geodesic from q to p instead of $|pq|$, plus the clockwise or counterclockwise distance to v_0 .

Note that a value of $\text{cw-dist}(q)$ or $\text{ccw-dist}(q)$ does not necessarily represent a unique position of q anymore: when q traverses an edge of P and the geodesic from q to p is along this edge in the opposite direction, $\text{cw-dist}(q)$ and $\text{ccw-dist}(q)$ do not change in value. However, it is sufficient to consider only the location of q that gives the shortest feed-link (if any such feed-link is optimal, then the shortest one is optimal too). All other adaptations to the algorithms are straightforward.

Theorem 3.6. *Given the boundary P of a simple polygon with n vertices, a set of obstacles with b vertices in total, and a point p inside P , we can compute the feed-link that minimizes the dilation from p to any point on P in $O(\lambda_7(nb) \log(nb))$ time.*

3.3 ... in a convex polygon

In the most simple case, where P is the boundary of a convex polygon, we can give a linear-time approximation scheme for placing one feed-link. This approximation algorithm computes one feed-link for which the dilation is at most a factor $1 + \varepsilon$ higher than for the optimal feed-link, for any constant $\varepsilon > 0$, and runs in $O(n + (1/\varepsilon) \log(1/\varepsilon))$ time. The idea is to choose a set Q of $\lceil 6\pi/\varepsilon \rceil$ points on P , angularly equal-spaced around p , and apply the algorithmic result of Theorem 3.3. The fact that one of the points of Q gives the desired approximation is shown in the next lemma.

Lemma 3.7. *Let q^* be the point on the boundary P of a convex polygon such that the feed-link to q^* realizes the minimum worst dilation δ^* . Then at least one point of Q can be used for a feed-link which gives a worst dilation of at most $(1 + \varepsilon)\delta^*$.*

Proof. Assume that q^* lies on an edge e , and the angle made at q^* when going from p straight to q^* and then clockwise on P is at most $\pi/2$. The other case is symmetric, and if q^* lies on a vertex then the proof is analogous as well. Let q be the first point of Q encountered when going clockwise on P from q^* . Then $\angle ppq^* \leq \varepsilon/3$ and $\angle pq^*q \leq \pi/2$.

Let r be any point on P , and compare the cases where $\overline{pq^*}$ is the only feed-link and where \overline{pq} is the only feed-link. We observe that the path-length from r to p is longer by at most $\mu(q^*, q) + |pq| - |pq^*|$ when we have \overline{pq} as the feed-link instead of $\overline{pq^*}$. Next we observe that $\mu(q^*, q) + |pq| \leq (\tan(\varepsilon/3) + 1/\cos(\varepsilon/3))|pq^*|$, because $(\mu(q^*, q) + |pq|)/|pq^*|$ is largest when q and q^* lie on the same edge of P and $\angle pq^*q = \pi/2$. We have

$$\tan(\varepsilon/3) + \frac{1}{\cos(\varepsilon/3)} = \frac{1 + \sin(\varepsilon/3)}{\cos(\varepsilon/3)} \leq \frac{1 + \varepsilon/3}{1 - \varepsilon/3} = 1 + \frac{2\varepsilon/3}{1 - \varepsilon/3} \leq 1 + \varepsilon,$$

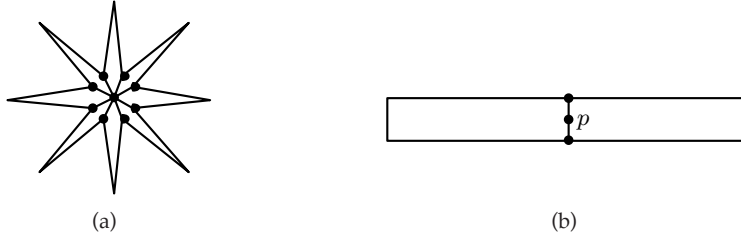


Figure 9: Dilation in a (a) non-convex polygon and (b) convex polygon. The solid lines inside the polygons show a smallest set of feed-links to guarantee constant dilation.

where the last inequality holds if $\varepsilon \leq 1$. Since the path-length from r when $\overline{pq^*}$ is the only feed-link is at least $|pq^*|$, we have

$$\frac{\delta_q(r)}{\delta_{q^*}(r)} = \frac{|pq| + |\text{sp}(r, q)|}{|pq^*| + |\text{sp}(r, q^*)|} \leq \frac{(1 + \varepsilon)|pq^*| + |\text{sp}(r, q^*)|}{|pq^*| + |\text{sp}(r, q^*)|} \leq 1 + \varepsilon,$$

and the lemma follows. \square

Theorem 3.8. *For any $\varepsilon > 0$, given the boundary P of a convex polygon with n vertices and a point p inside it, we can compute a feed-link that minimizes the dilation within a factor $1 + \varepsilon$ of the optimal dilation in $O(n + (1/\varepsilon) \log(1/\varepsilon))$ time.*

4 Placing multiple feed-links . . .

If we place only one feed-link from point p to the boundary P of a simple polygon, the resulting dilation can still be large even if we optimize the placement. In practice we would always want the dilation to be below a certain constant value. This motivates the study of placing multiple feed-links and, in particular, the problem: given a maximum allowed dilation, place as few feed-links as possible from p so that the dilation is at most the given maximum. Although an exact solution appears difficult, we present a simple algorithm in Subsection 4.1 that places at most one feed-link too many in linear time.

Figure 9(a) shows that some simple polygons require many feed-links to realize a dilation below a certain value; $n/2$ feed-links may even be necessary to ensure a dilation below any constant, no matter how large the constant. On the other hand, for convex polygons, we will show in Subsection 4.2 that two feed-links are always sufficient and sometimes necessary to realize a constant dilation. Furthermore, with k feed-links we can even guarantee a dilation as low as $1 + O(1/k)$.

Because simple polygons are too general and convex polygons are too specific, we also study a class of realistic polygons (to be precise, (α, β) -covered polygons [14]) in Subsection 4.3, and show that a constant number of feed-links are sufficient to realize a constant dilation.

4.1 . . . in a simple polygon

In this section we study the computational problem of, given a simple polygon P , a point p inside P and a target dilation $c > 1$, finding a (small) set of feed-links that connect p to P to

guarantee a dilation of at most c . We give a simple algorithm that finds a set of feed-links that contains at most one feed-link more than a smallest set.

The algorithm proceeds in a greedy fashion. We start by choosing an arbitrary first feed-link $\overline{pq_1}$. For instance, we might choose the point q_1 on P closest to p . We want to place the next feed-link q_2 (in clockwise order) as far from q_1 as possible such that all points in between q_1 and q_2 have a detour (via q_1 or q_2) not larger than c .

We first traverse P starting at q_1 until we reach a point m_1 for which another point just beyond m_1 has a detour larger than c via q_1 , as in Figure 3(c). To find the point m_1 we traverse the edges of P clockwise, starting at q_1 . As in Subsection 2.2 we check for a maximum of the dilation on the edge, and whether it is larger than c . If it is, we find the point m_1 where detour reaches c by parametrizing points on the edge by $t \in [0, 1]$ and solving the corresponding quadratic equation in t .

Let us make some observations about point q_2 on P . The point q_2 is maximal (in the sense of being furthest from m_1 in clockwise direction) with the property that all points in between m_1 and q_2 have detour not larger than c via q_2 . Let ℓ be the network distance from m_1 to p via q_1 . Then q_2 cannot be placed further than the point \hat{q} for which the network distance from m_1 to p via \hat{q} equals ℓ , as well. This is because the dilation between p and m_1 via \hat{q} is, by definition, c . Thus any position for q_2 further than \hat{q} would result in a point with dilation strictly higher than c . However, we may have to place q_2 closer to q_1 than \hat{q} , since it is possible that some point between m_1 and \hat{q} still has a higher detour.

In more detail, we will traverse the edges of P in clockwise order, starting at m_1 , and maintain the maximum network distance ℓ_{\max} from m_1 via q_2 to p that is allowed while ensuring a dilation of at most c for all points between m_1 and q_2 . Initially, $\ell_{\max} = \ell$. We also maintain the boundary distance d that has been traversed from m_1 during the search for the location of q_2 .

Consider the next edge e in the clockwise traversal.

1. We parametrize a point $e(t)$ on e , determine the smallest maximum allowed network distance ℓ' for all points on e by stating the dilation of $e(t)$ expressed in terms of ℓ' and t , and maximizing it. This yields a value of t and therefore a point on e that realizes the maximum dilation. We know that the maximum dilation should be at most c , so we can compute an upper bound on ℓ' . If $\ell' < \ell_{\max}$, we set $\ell_{\max} = \ell'$.
2. We test if we must place q_2 on e using the known values of d and ℓ_{\max} , again by parameterizing a point $e(t)$ on e . If so, we place q_2 at the appropriate location, otherwise we update d with the length of e and proceed with the next clockwise edge after e .

After we placed q_2 , we keep on traversing the boundary of P clockwise until we encounter the last point that can still use the feed-link at q_2 and have dilation at most c . This is where we place m_2 . The process continues until we have traversed the whole boundary of P (when we reach q_1).

Theorem 4.1. *Given the boundary P of a simple polygon with n vertices, a point p inside P , and a maximum dilation value c , an $O(n + k)$ time algorithm exists that determines $k + 1$ feed-links such that the dilation is at most c everywhere on P , where k is the minimum number of feed-links needed to achieve dilation c . If P contains obstacles with b vertices in total, the algorithm runs in $O(nb \log(nb) + k)$ time.*

Proof. Assume the algorithm places $k + 1$ feed-links at q_1, \dots, q_{k+1} ($k \geq 0$). By definition of the q_i , any set of feed-links achieving a dilation at most c needs to have a feed-link in each

of the k sectors between q_i and q_{i+1} for $i = 1, \dots, k$, a feed-link between q_{k+1} and q_1 may not be needed.

For polygons without obstacles the running time of the algorithm is $O(n + k)$ since it spends constant time per edge of the polygon with an additional overhead of at most $O(k)$ for placing the feed-links. For polygons with obstacles we use the same approach as described in Section 2.2. That is, we compute the subdivision S in $O(b \log b)$ time and overlay it with P in $O(nb \log(nb))$ time. The overlay yields $O(nb)$ edges on P . We then run the algorithm as before. \square

4.2 ... in a convex polygon

Let P be a convex polygon and let p be a point inside P . We explore how many feed-links are necessary and sufficient to guarantee constant detour for all points on P .

One feed-link is not sufficient to guarantee constant dilation. Consider a rectangle with width w and height $h < w$, and let p be its center, as illustrated in Figure 9(b). Using one feed-link, the midpoint of one of the long sides will have detour greater than $2w/h$, which can be arbitrarily large. Hence two feed-links may be necessary.

Two feed-links are also sufficient to guarantee constant dilation for all points on P . In fact we argue that we can always choose two feed-links such that the dilation is at most $3 + \sqrt{3} \approx 4.73$. This bound is not far from the optimum, since an equilateral triangle with p placed in the center has dilation at least $2 + \sqrt{3} \approx 3.73$ for any two feed-links. To see that, observe that one of the sides of the equilateral triangle does not have a feed-link attached to it (or only at a vertex), which causes the point at the middle of that side to have detour at least $2 + \sqrt{3}$.

Let q be the closest point to p on P . We choose \overline{pq} as the first feed-link. Consider the smallest equilateral triangle Δ that contains P and that is oriented such that one of its edges contains q . Let e_0 be the edge of Δ containing q , and let e_1 and e_2 be the other edges, in clockwise order from e_0 (see Figure 10(a)). By construction, each edge of Δ is in contact with P . Let t_1 be a point of P in contact with e_1 , and let t_2 be a point of P in contact with e_2 . Let q' be the point on $P[t_1, t_2]$ that is closest to p among the points on $P[t_1, t_2]$. We choose $\overline{pq'}$ as the second feed-link. In Appendix A we prove that these two feed-links guarantee a dilation of at most $3 + \sqrt{3}$.

Theorem 4.2. *Given the boundary P of a convex polygon and a point p inside it, two feed-links from p to P are sufficient to achieve a dilation of $3 + \sqrt{3}$. The feed-links can be computed in linear time.*

We now consider the general setting of placing k feed-links, where k is a constant larger than 1. We prove that placing the feed-links at an equal angular distance of $\eta = 2\pi/k$ guarantees a dilation of $1 + O(1/k)$. To simplify the argument we choose $k \geq 6$ (the result for smaller k immediately follows from the result for two feed-links). Our proof is based on the following lemma.

Lemma 4.3. *Let q_1 and q_2 be two points on the boundary P of a convex polygon such that the angle $\angle q_1 p q_2 = \eta \leq \pi/3$, and let $\overline{pq_1}$ and $\overline{pq_2}$ be feed-links. Then for all points $r \in P[q_1, q_2]$, we have $\delta_{q_1, q_2}(r) \leq 1 + \eta$.*

Proof. Given $r \in P[q_1, q_2]$, let t_1 and t_2 be the intersection points of a tangent line of P at r and the supporting lines of $\overline{pq_1}$ and $\overline{pq_2}$, respectively (see Figure 10(b)). Let



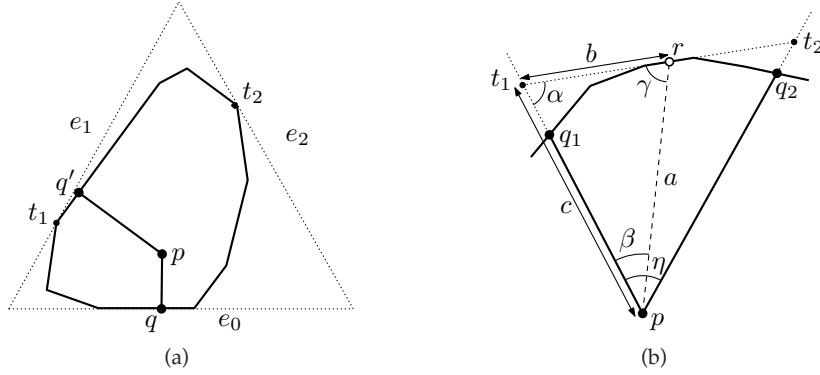


Figure 10: (a) Placement of two feed-links for a convex polygon that realize a constant dilation. (b) Illustration for Lemma 4.3: $(b + c)/a$ bounds the detour at r .

$\alpha = \max\{\angle pt_1r, \angle pt_2r\}$. Since $\angle q_1pq_2 = \eta$, we have $\frac{\pi-\eta}{2} \leq \alpha \leq \pi - \eta$. By symmetry we can assume $\angle pt_1r = \alpha$. Let $\angle rpt_1 = \beta$, $\angle t_1rp = \gamma$, $a = |pr|$, $b = |t_1r|$, and $c = |pt_1|$ (see Figure 10(b)). Then $0 \leq \beta \leq \eta$ and $0 \leq \gamma \leq \frac{\pi+\eta}{2}$.

We now observe that $\delta_{q_1, q_2}(r) \leq \delta_{q_1}(r) \leq (b+c)/a$. We note that for any given β , $(b+c)/a$ is maximized if α is minimized, so we can assume $\alpha = (\pi - \eta)/2$. Next we use Newton's formula, i.e.,

$$\frac{b+c}{a} = \frac{\cos((\beta-\gamma)/2)}{\sin(\alpha/2)}.$$

We first bound $(\beta-\gamma)/2$. We have $\beta+\gamma = \pi-\alpha = (\pi+\eta)/2$ and $0 \leq \beta \leq \eta$. From this we get that $(\beta-\gamma)/2 = (2\beta-(\beta+\gamma))/2 = \beta-(\pi+\eta)/4$, thus $(\beta-\gamma)/2$ is minimized if $\beta = 0$ and maximized if $\beta = \eta$. Therefore, we have the bounds $-(\pi+\eta)/4 \leq (\beta-\gamma)/2 \leq (3\eta-\pi)/4$. Since $\eta \leq \pi/3$,

$$-\frac{\pi}{3} \leq \frac{\beta-\gamma}{2} \leq 0.$$

Thus, $\cos((\beta-\gamma)/2)$ is maximized if $(\beta-\gamma)/2$ is maximized, i.e., if $\beta = \eta$ and $\gamma = (\pi-\eta)/2 = \alpha$. Thus we have $a = c$, $b = 2 \sin(\eta/2)a$, and $(b+c)/a = 1 + 2 \sin(\eta/2) < 1 + \eta$. \square

Setting $\eta = 2\pi/k$ in Lemma 4.3, we obtain the following result.

Theorem 4.4. *Given the boundary P of a convex polygon and a point p inside it, k feed-links from p to P are sufficient to achieve a dilation of $1 + O(1/k)$.*

4.3 ... in a realistic polygon

Even though the result of the previous section is not true for general simple polygons, intuitively a constant number of feed-links should guarantee constant dilation for *realistic* polygons. Therefore, we define a class of simple polygons to be *feed-link realistic* if there are two constants $\delta > 1$ and $c \geq 1$, such that for every polygon P in the class and every point p in the interior of P , there exist c feed-links that achieve a dilation of at most δ for any point on the boundary of P . Many different classes of realistic polygons have been suggested in

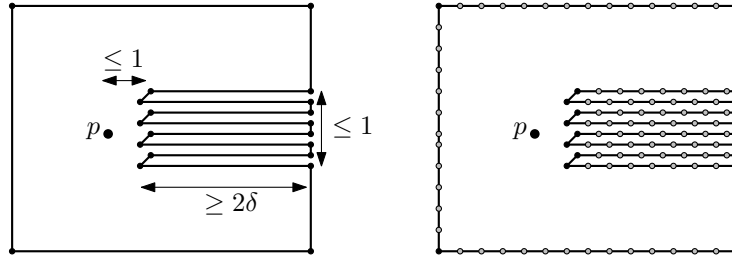


Figure 11: A β -fat polygon (left) and an adaptation (right) that require many feed-links.

the literature. We show that most of them do not imply feed-link realism, but that one of them does.

Consider the left polygon in Figure 11. At least c feed-links are required to obtain a dilation smaller than δ , if the number of prongs is c and their length is at least δ times larger than the distance of their leftmost vertex to p . No feed-link can give a dilation at most δ for the leftmost vertex of more than one dent. However, the polygon is β -fat [11].

Definitions that depend on the spacing between the vertices or edge-vertex distances will also not give feed-link realism, because the left polygon in Figure 11 can be turned into a realistic polygon according to such definitions. We simply add extra vertices on the edges to get the right polygon: it has edge lengths that differ by a factor of at most 2, it has no vertex close to an edge in relation to the length of that edge, and it has no sharp angles. The extra vertices obviously have no effect on the dilation. This shows that definitions like low density (of the edges) [25], unclutteredness (of the edges) [9, 11], locality [15], and another fatness definition [26] cannot imply feed-link realism.

However, we can argue that polygons that are (α, β) -covered [14] are feed-link realistic. For an angle ϕ and a distance d , a (ϕ, d) -triangle is a triangle with all angles at least ϕ and all edge lengths at least d . Let P be the boundary of a simple polygon, let $\text{diam}(P)$ be the diameter of P , and let $0 < \alpha < \pi/3$ and $0 < \beta < 1$ be two constants. P is (α, β) -covered if for each point on P , an $(\alpha, \beta \cdot \text{diam}(P))$ -triangle exists with a vertex at that point, whose interior is completely inside P [14]. The proof of the following theorem, which shows that $O(1)$ feed-links suffice for an (α, β) -covered polygon, is given in Appendix B.

Theorem 4.5. *Given the boundary P of an (α, β) -covered polygon and a point p inside it, $\frac{4\pi}{\alpha}$ feed-links are sufficient to achieve a dilation of $\frac{4\pi c}{\beta^2 \alpha \sin \alpha \sin \frac{1}{2}\alpha}$, for some absolute constant $c > 0$.*

5 Heuristics and experiments

Intuitively, even simple heuristics to place feed-links may work well to realize a low dilation in many reasonable cases. In this section we investigate this intuition by presenting such heuristics and testing them on simple polygons with obstacles. The heuristics can all be implemented to run in linear time for placing any constant number of feed-links. Hence, even for one feed-link the heuristics are more efficient than the optimal algorithm.



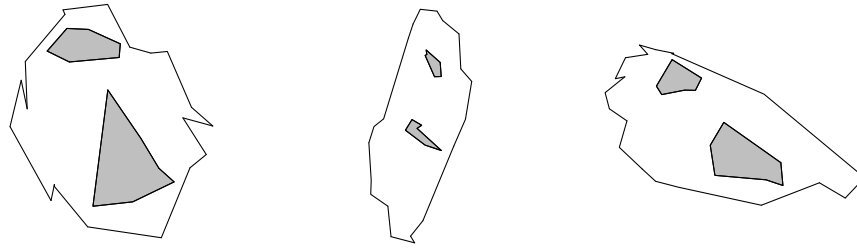


Figure 12: Randomly generated polygons with two obstacles.

5.1 Heuristics for placing multiple feed-links

The first heuristic is called the *greedy heuristic*. If only one feed-link is required, we connect the feed-link to the closest point on P , taking the obstacles into account. If more than one feed-link is required, then we first use the feed-link to the closest point, then we iteratively find the point with worst detour so far, and add a feed-link to this point. We repeat this until we have the required number of feed-links.

The other two heuristics use sectors rooted at p . Assume we wish to place k feed-links. We construct a collection of k equally spaced half-lines emanating from p . In every sector, we determine the point on P that is closest to p , i.e., has the shortest path to p avoiding obstacles, and make a feed-link to that point.

The sector approach does not specify a base angle for the k half-lines, that is, they can be rotated around p as a whole and still give sectors with equal opening angles. We may just choose a base angle at random. We may also place the half-lines after computing the closest point, and choose the base angle such that the closest point is in the middle of a sector. We call these two methods the *random* and the *positioned sector heuristics*, respectively. The positioned heuristic makes sure, for instance, that when placing only two feed-links to points q_1 and q_2 , the angle $\angle q_1 p q_2 \geq \pi/k = \pi/2$, whereas for the random heuristic this angle can be arbitrarily small, and q_1 and q_2 may be very close.

Theorem 5.1. *For boundary P of a polygon with n vertices without obstacles inside, the greedy heuristic and the sector heuristic take $O(n)$ time to find $k = O(1)$ feed-links, where n is the number of vertices of P . For a polygon boundary P with obstacles inside, the greedy heuristic and the sector heuristic run in $O(nb \log(nb))$ time, where b is the number of vertices of the obstacles.*

We note that it is unlikely in practice that the heuristics require $O(nb)$ time for a polygon with obstacles. The overlay of the shortest path map and the polygon can be expected to have much fewer vertices than $O(nb)$, and these vertices can be computed in an output-sensitive manner (for example by a plane sweep algorithm).

5.2 Experimental results

We have implemented the three heuristics described in the previous section, and have run tests on a collection of 100 polygons with 2 obstacles. The objective is to determine the dilation of typical polygons one can expect in practice when using 1, 2, or more feed-links, and to compare which feed-link placement heuristic gives the best resulting dilation.

First we implemented a polygon-and-obstacles generator and adjusted it to generate a collection that (visually) seems reasonable for the application at hand: incomplete road

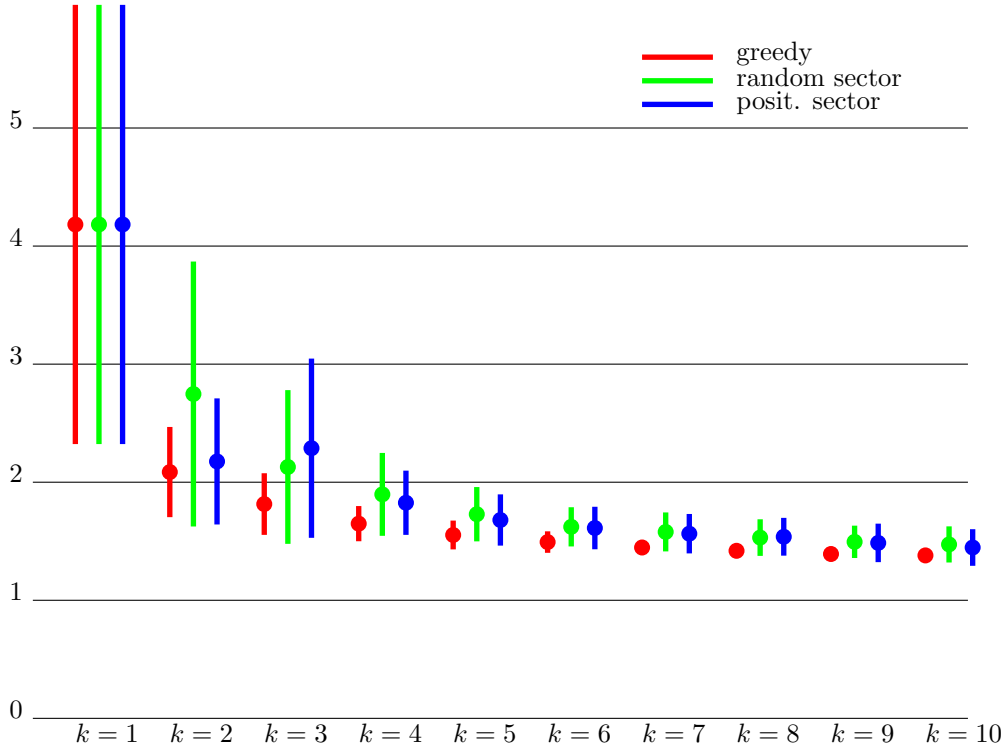


Figure 13: The mean μ and standard deviation σ of the dilation values for k feed-links.

networks. Also for this reason, we did not consider more than two obstacles in the polygon. The generator starts by creating a triangle from three random points, and then iterates until the polygon has 20 vertices. Based on a tuning parameter, the algorithm extends the polygon by either a) inserting a new vertex at a random point on the polygon boundary, moving the point perpendicular to the edge and along the tangent based on two more parameters; or b) selecting two vertices, adding the line segment between them as a new edge if it lies completely outside of the polygon, while removing the vertices in the part of the polygon that is closed off by this new line segment. Once the boundary polygon is generated, obstacles are added by a similar procedure which generates a triangle within the polygon, and adds vertices and edges if these do not intersect other obstacles or the boundary. Figure 12 shows various polygons that were generated. We chose a random point p in each generated polygon, and ran the heuristics for $1, \dots, 10$ feed-links. The heuristics were implemented in Java, without the use of third-party libraries. We used Dijkstra's algorithm on the visibility graph for the distance calculations required to compute the dilation. Figure 13 shows the results.

For one feed-link, all three heuristics will choose the same feed-link, so the results are the same. For more feed-links, it appears that the greedy heuristic outperforms the other two. Figure 14 shows four examples of the greedy heuristic, run on the same polygon for different numbers of feed-links. The two sector heuristics perform comparably with respect to each other, although the positioned sector heuristic seems to work better for two

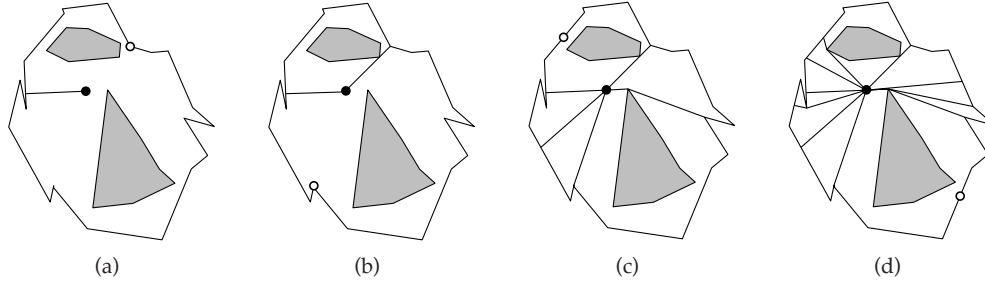


Figure 14: Four examples of the greedy heuristic with different numbers of feed-links. The circle indicates the point with the worst detour. (a) One feed-link, dilation is 4.0513. (b) Two feed-links, dilation is 2.5581. (c) Five feed-links, dilation is 1.7753. (d) Ten feed-links, dilation is 1.3137.

feed-links and the random sector heuristic seems to work better for three feed-links. We notice that the (average) dilation goes down with more feed-links, which is to be expected. Already for three feed-links, the dilation obtained by the greedy heuristic is below 2 on the average.

k	lower bound	upper bound
1	3.7744	3.7753
2	1.8758	1.8776
3	1.5710	1.5956
4	1.3862	1.4854

Table 1: Bounds on the average optimal dilation μ_{opt} .

Figure 13 shows how the three heuristics compare to each other, but does not show how close they get to the best achievable dilation. To determine this, we have implemented methods that can approximate upper and lower bounds on the maximum dilation with a certain precision, given P , p , and the number of feed-links k . These methods sample many points on the boundary and try all combinations of feed-links between the sampled points and p . In combination with the known maximum perimeter length that does not have a sampled point, we obtain upper and lower bounds on the maximum dilation. This method becomes computationally too demanding when there are more than four feed-links (or the lower and upper bounds start to differ too much). The results for $k = 1, \dots, 4$ are given in Table 1. We observe that the greedy heuristic is about 10–20% off the optimum dilation, for $k = 1, \dots, 4$. For more feed-links, we do not know this.

6 Conclusions

We studied the problem of extending a partial road network by adding feed-links to relevant disconnected locations. For proximity analysis in GIS, this is often necessary to make sure that all relevant locations are reachable via the road network [6, 8, 12, 21]. Previous

work makes no attempt to quantify the quality of such feed-links. In this paper, we propose to make such a quantification using the concept of *dilation*.

We presented an efficient algorithm to compute the dilation of a set of feed-links that connect a point to a simple polygon boundary, and an algorithm to compute one feed-link while minimizing the dilation obtained. We showed that, for a given dilation, we can place $k + 1$ feed-links to realize this dilation when the minimum possible is k (we may place one feed-link too many). These results also apply to polygons with obstacles, although the worst-case running time of the algorithms is larger. Furthermore, we showed that two feed-links are sometimes necessary and always sufficient to guarantee constant dilation for convex polygons. By placing k feed-links, we can even guarantee a dilation of at most $1 + O(1/k)$. Finally, we considered the number of feed-links necessary for realistic polygons, and proved that (α, β) -covered polygons require only a constant number of feed-links for constant dilation. For other definitions of realistic polygons such a result provably does not hold. Finally, we did an experimental study on the dilation that can be achieved using feed-links in randomly generated “realistic” polygons.

A number of interesting and challenging extensions of our work are possible. We mention a few, but several other possibilities exist as well. Firstly, identifying an optimal placement for more than one feed-link seems difficult, but would be of interest to solve. Secondly, we did not consider the situation where several points lie inside P and need to be connected via feed-links. Here we may or may not want to allow one feed-link to connect to another feed-link. Thirdly, we could define the optimal feed-link to be the one that minimizes the average dilation instead of the maximum dilation. Finally, assume we are given an incomplete road network N and several locations, which can lie in different faces of the graph induced by N . How should we place optimal feed-links for all disconnected locations in this setting? This question is the actual problem as it occurs in GIS context. Since our objective was to define optimality of feed-links and obtain provable results, we considered a fairly restricted version of the original problem in this paper. It would be of interest to address the general problem and obtain provable results as well.

Acknowledgments

The authors thank Mark de Berg for pointing us to the results in [5]. This research has been supported by the Netherlands Organisation for Scientific Research (NWO) under BRICKS/FOCUS grant number 642.065.503, under the project GOGO, and under project no. 639.022.707. B. Aronov has been partially supported by a grant from the U.S.-Israel Binational Science Foundation, by NSA MSP Grants H98230-06-1-0016 and H98230-10-1-0210, and NSF Grant CCF-08-30691. M. Buchin is supported by the German Research Foundation (DFG) under grant number BU 2419/1-1. M. Löffler has been partially supported by the U.S. Office of Naval Research under grant N00014-08-1-1015. R. I. Silveira has been supported by the Netherlands Organisation for Scientific Research (NWO).

References

- [1] AGARWAL, P., KLEIN, R., KNAUER, C., LANGERMAN, S., MORIN, P., SHARIR, M., AND SOSS, M. Computing the detour and spanning ratio of paths, trees,

- and cycles in 2D and 3D. *Discrete & Computational Geometry* 39, 1 (2008), 17–37. doi:10.1007/s00454-007-9019-9.
- [2] AGARWAL, P., SHARIR, M., AND SHOR, P. Sharp upper and lower bounds on the length of general Davenport-Schinzel sequences. *Journal of Combinatorial Theory, Series A* 52, 228–274 (1989).
 - [3] ARONOV, B., BUCHIN, K., BUCHIN, M., JANSEN, B., DE JONG, T., VAN KREVELD, M., LÖFFLER, M., LUO, J., SILVEIRA, R., AND SPECKMANN, B. Feed-links for network extensions. In *Proc. 16th International Symposium on Advances in Geographic Information Systems* (2008), pp. 308–316. doi:10.1145/1463434.1463478.
 - [4] ARONOV, B., BUCHIN, K., BUCHIN, M., VAN KREVELD, M., LÖFFLER, M., LUO, J., SILVEIRA, R., AND SPECKMANN, B. Connect the dot: Computing feed-links with minimum dilation. In *Proc. Algorithms and Data Structures, 11th International Symposium (WADS)*, vol. 5664 of *Lecture Notes in Computer Science*. Springer, Berlin, 2009, pp. 49–60. doi:10.1007/978-3-642-03367-4_5.
 - [5] BOSE, P., CHEONG, O., AND DUJMOVIĆ, V. A note on the perimeter of fat objects. *Computational Geometry: Theory and Applications* 44 (2011), 1–8. doi:10.1016/j.comgeo.2010.06.002.
 - [6] DAHLGREN, A. A proximity analysis application for large source dataset. In *Proc. of ScanGIS 2005* (2005), pp. 125–136.
 - [7] DAHLGREN, A., AND HARRIE, L. Evaluation of computational methods for connecting points to large networks. *Mapping and Image Science* 4 (2006), 45–54.
 - [8] DAHLGREN, A., AND HARRIE, L. The development of a tool for proximity analysis. In *Proc. 10th AGILE International Conference on Geographic Information Science* (2007).
 - [9] DE BERG, M. Linear size binary space partitions for uncluttered scenes. *Algorithmica* 28 (2000), 353–366. doi:10.1007/s004530010047.
 - [10] DE BERG, M., CHEONG, O., VAN KREVELD, M., AND OVERMARS, M. *Computational Geometry: Algorithms and Applications*, third ed. Springer, Berlin, 2008.
 - [11] DE BERG, M., VAN DER STAPPEN, A., VLEUGELS, J., AND KATZ, M. Realistic input models for geometric algorithms. *Algorithmica* 34 (2002), 81–97. doi:10.1007/s00453-002-0961-x.
 - [12] DE JONG, T., , AND TILLEMA, T. Transport network extensions for accessibility analysis in geographic information systems. In *Proc. AfricaGIS* (2005).
 - [13] EBBERS-BAUMANN, A., GRÜNE, A., AND KLEIN, R. The geometric dilation of finite point sets. *Algorithmica* 44 (2006), 137–149. doi:10.1007/s00453-005-1203-9.
 - [14] EFRAT, A. The complexity of the union of (α, β) -covered objects. *SIAM Journal on Computing* 34 (2005), 775–787. doi:10.1137/S0097539702407515.
 - [15] ERICKSON, J. Local polyhedra and geometric graphs. *Computational Geometry: Theory and Applications* 31 (2005), 101–125. doi:10.1016/j.comgeo.2004.08.004.

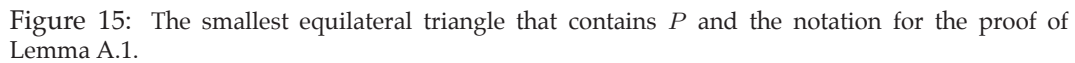
- [16] FARSHI, M., GIANNOPOULOS, P., AND GUDMUNDSSON, J. Improving the stretch factor of a geometric network by edge augmentation. *SIAM Journal on Computing* 38 (2008), 226–240. doi:10.1137/050635675.
- [17] HERSHBERGER, J. Finding the upper envelope of n line segments in $O(n \log n)$ time. *Information Processing Letters* 33 (1989), 169–174. doi:10.1016/0020-0190(89)90136-1.
- [18] HERSHBERGER, J., AND SURI, S. An optimal algorithm for Euclidean shortest paths in the plane. *SIAM Journal on Computing* 28, 6 (1999), 2215–2256. doi:10.1137/S0097539795289604.
- [19] NARASIMHAN, G., AND SMID, M. Approximating the stretch factor of Euclidean graphs. *SIAM Journal on Computing* 30, 978–989 (2000). doi:10.1137/S0097539799361671.
- [20] NARASIMHAN, G., AND SMID, M. *Geometric Spanner Networks*. Cambridge University Press, Cambridge, UK, 2007.
- [21] NESS, B., AND BROGAARD, S. GIS proximity analysis and environmental assessment of sugar beet transport in Scania, Sweden. *Area* 40 (2008), 459–471.
- [22] NORDBECK, S. Computing distances in road nets. *Papers in Regional Science* 12 (1964), 207–220.
- [23] RITSEMA VAN ECK, R. *Analyse van Transportnetwerken in GIS voor Sociaal-geografisch Onderzoek*. PhD thesis, Utrecht University, 1993.
- [24] SHARIR, M., AND AGARWAL, P. *Davenport-Schinzel Sequences and Their Geometric Applications*. Cambridge University Press, 1995.
- [25] VAN DER STAPPEN, A., OVERMARS, M., DE BERG, M., AND VLEUGELS, J. Motion planning in environments with low obstacle density. *Discrete & Computational Geometry* 20 (1998), 561–587. doi:10.1007/PL00009402.
- [26] VAN KREVELD, M. On fat partitioning, fat covering and the union size of polygons. *Computational Geometry: Theory and Applications* 9 (1998), 197–210. doi:10.1016/S0925-7721(96)00016-8.

A Two feed-links in convex polygons

In this appendix, we will prove Theorem 4.2. Recall that q is the closest point to p on P ; that Δ is the smallest equilateral triangle that contains P and that touches P in q ; that t_1 and t_2 are points of P on the other two edges of Δ ; and that q' is the point on $P[t_1, t_2]$ that is closest to p . Refer to Figure 15. We will separately bound the dilation from the points on $P[t_2, t_1]$ via q and the dilation from the points on $P[t_1, t_2]$ via q' .

Lemma A.1. *For any point $r \in P[t_2, t_1]$, $\delta_q(r) \leq 3 + \sqrt{3}$.*

Proof. Assume without loss of generality that $r \in P[t_2, q]$, and that r is not a vertex (vertices can be analyzed in a similar way); see Figure 15. Let ℓ be the line tangent to the polygon at r . To bound the detour at r , we first note that because the interior of P is convex and



To bound l_1 and l_2 , assume that the origin is at p , and let the coordinates of r be (x, y) . We consider the case where $x, y \geq 0$; the other cases are symmetric. We need to compute two of the sides of the triangle shown shaded in Figure 15. The height of the triangle is $|pq| + y$. Because the angle at s is $\pi/3$, we get

Therefore we can bound the detour at r :

Using $|pq| \leq |pr|$ and $(x + \sqrt{3}y) \leq 2|pr|$ (which can be proved), we obtain

Lemma A.2 can be proven with the same arguments as Lemma A.1. From Lemmas A.1 and A.2 we conclude:

JOSIS, Number 3 (2011), pp. 3–31

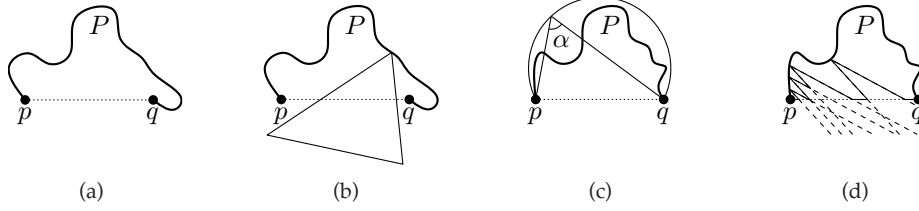


Figure 16: (a) Two points that see each other. (b) If $P[p, q]$ is short, all triangles intersect \overline{pq} . (c) $P[p, q]$ must stay inside the circle of points that have angle α with p and q . (d) The fat path marks the upper boundary of the union of all d -directional cones.

B Feed-links in realistic polygons

In this appendix we will show that for (α, β) -covered polygons, a constant number of feed-links suffices to obtain a constant dilation, and prove Theorem 4.5. First, though, we will prove the same result (but with a better dependency on α and β) for a more specific class of polygons, which we call (α, β) -immersed.

Recall that, for an angle ϕ and a distance d , a (ϕ, d) -triangle is a triangle with all angles at least ϕ and all edge lengths at least d . Let P be a simple polygon boundary, let $\text{diam}(P)$ be the diameter of P , and let $0 < \alpha < \pi/3$ and $0 < \beta < 1$ be two constants. P is (α, β) -covered if, for each point on P , an $(\alpha, \beta \cdot \text{diam}(P))$ -triangle exists with a vertex at that point, whose interior is completely inside P [14]. Furthermore, P is (α, β) -immersed if for each point on P there is such a triangle completely inside or on P and one completely outside or on P . Note that an (α, β) -immersed polygon boundary is also (α, β) -covered. For ease of description, we assume in the remainder that $\text{diam}(P) = 1$.

We will use a result by Bose et al. [5] that bounds the perimeter of P as a function of α and β :

Lemma B.1 (Bose et al.). *The perimeter of an (α, β) -covered polygon is at most $\frac{c}{\beta \sin \alpha}$, for some absolute constant $c > 0$.*

Also, we need a technical lemma that states that if the distance between two points on P along P is short enough, then it is proportional to the Euclidean distance.

Lemma B.2. *If p and q can see each other on the inside of an (α, β) -covered polygon with boundary P and $\mu(p, q) < \beta$, then $\mu(p, q) < f(\alpha) \cdot |pq|$, where $f(\alpha) \leq \frac{2\pi}{\alpha \sin \frac{1}{4}\alpha}$.*

Proof. Assume that q is to the right of p on a horizontal line, as in Figure 16(a). They see each other, so \overline{pq} is inside P . We are interested in $P[p, q]$.

Since P is (α, β) -covered, each point on $P[p, q]$ has an (α, β) -triangle inside P . The lengths of the sides of these triangles are at least β . Therefore, since $\mu(p, q) < \beta$, no such side can lie completely in the part of P above \overline{pq} , see Figure 16(b). This means that for each point r on $P[p, q]$, there must be an empty cone of angle α that lies between p and q . This immediately implies that $P[p, q]$ must stay inside a circle through p and q with radius R , see Figure 16(c). Here $R = \frac{|pq|}{2 \sin \alpha}$.

To bound the length of $P[p, q]$, we argue as follows. For each point r on $P[p, q]$, there is at least one empty cone of angle $\frac{1}{2}\alpha$ that is aligned with a radial grid of directions: for

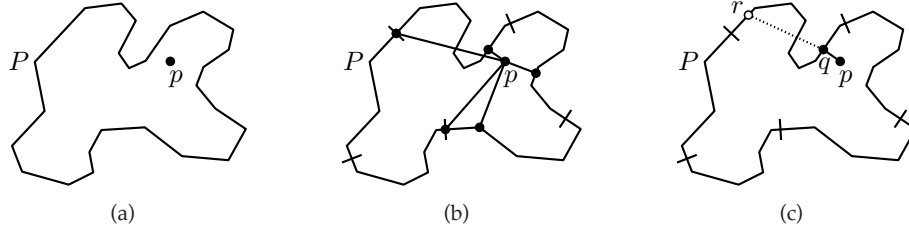


Figure 17: (a) A polygon P that is (α, β) -immersed. (b) A feed-link to the closest point on each boundary portion of length β . (c) The detour of r is constant, because the boundary distance between r and q is bounded by their Euclidean distance.

$0 < i < \frac{4\pi}{\alpha}$, let r_i be the direction with angle $i \cdot \frac{\alpha}{2}$ with respect to the x -axis. The union of these cones for all points r has $P[p, q]$ as part of its boundary (but there may be additional obstacles and parts below \overline{pq}).

For each i , consider the union of the (possibly infinite number of) cones in direction i , see Figure 16(d). We want to bound the length of the upper boundary of these cones inside P above \overline{pq} . Such a path is monotone in the direction perpendicular to d_i , and can have only a limited steepness, so if it travels distance ξ in that direction it will be at most $\frac{\xi}{2 \sin \frac{1}{4}\alpha}$ long. A rough upper bound is $\frac{|pq|}{2 \sin \frac{1}{4}\alpha}$. The length of $P[p, q]$ is now bounded by the sum of these paths over all values of i . This gives a bound of $\frac{4\pi|pq|}{2\alpha \sin \frac{1}{4}\alpha}$, as claimed. \square

B.1 (α, β) -Immersed Polygons

When P is (α, β) -immersed, each point on the boundary has an empty (α, β) -triangle outside P as well as inside P . This implies that Lemma B.2 also holds for two points p and q that can see each other on the *outside* of the polygon.

Theorem B.3. *When P is (α, β) -immersed, we can place $\frac{c}{\beta^2 \sin \alpha}$ feed-links such that the dilation is at most $1 + \frac{4\pi}{\alpha \sin \frac{1}{4}\alpha}$, for some absolute constant $c > 0$.*

Proof. We give a constructive proof. Given an (α, β) -immersed polygon and a point p inside it, we split P into portions of length β . By Lemma B.1 there are at most $\frac{c}{\beta^2 \sin \alpha}$ portions. On each portion, we place a feed-link to the closest point to p . Figure 17(b) shows the resulting feed-links in an example.

For any point r on P , we show that the detour is constant. Consider the portion of P containing r , and the point q that is the closest point to p on that portion, as in Figure 17(c). The segment \overline{qr} may intersect P in a number of points. For each pair of consecutive intersection points, they can see each other either inside or outside P . Since P is (α, β) -immersed, Lemma B.2 applies to each pair, and hence $\mu(q, r) \leq f(\alpha) \cdot |qr|$. Also, we know that $|pq| \leq |pr|$. We conclude that the detour is bounded by

$$\begin{aligned} \delta_q(r) &= \frac{|pq| + \mu(q, r)}{|pr|} \leq \frac{|pq| + f(\alpha)|qr|}{|pr|} \\ &\leq \frac{|pq| + f(\alpha)(|pr| + |pq|)}{|pr|} \leq \frac{|pr| + f(\alpha)(|pr| + |pr|)}{|pr|} = 1 + 2f(\alpha). \end{aligned}$$

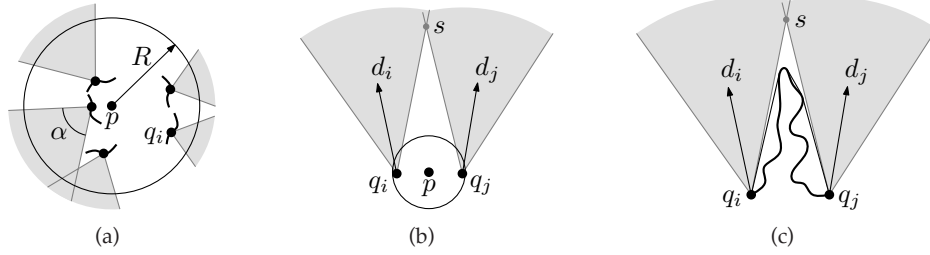


Figure 18: (a) A disk around p of radius R contains the points q_i such that $\overline{pq_i}$ is a feed-link. (b) If the angle between two bisecting directions d_i and d_j is small, the (α, β) -triangles intersect in s . (c) The boundary length between q_i and q_j cannot be too long.

□

B.2 (α, β) -Covered Polygons

When P is (α, β) -covered but not (α, β) -immersed, the proof no longer works since there can be two points that see each other outside the polygon, in which case Lemma B.2 does not hold. However, we can still prove that (α, β) -covered polygons are feed-link realistic, albeit with a worse dependence on α and β .

Let $C = \frac{4\pi c}{\beta^2 \alpha \sin \alpha \sin \frac{1}{2}\alpha}$, for an absolute constant c as in Lemma B.1. We incrementally place feed-links until the dilation is at most C everywhere. In particular, after placing the first i feed-links, consider the set of points on P that have detour worse than C . If q_{i+1} is the point of this set that is closest to p , then we let the next feed-link be $\overline{pq_{i+1}}$. We repeat this process until there are no points of P with detour exceeding C .

We now need to prove that this results in a constant number of feed-links. So, say we placed k feed-links this way, and let their points be $q_1 \dots q_k$. Obviously, we have $|pq_i| \leq |pq_j|$ if $i < j$.

Lemma B.4. *Unless $k = 1$, all points q_i are in a disk D centered at p of radius $R = \frac{1}{2}\beta \sin \frac{1}{2}\alpha$.*

Proof. After we place the first feed-link, every point of P is connected to p via some path. By Lemma B.1, $\mu(P) \leq \frac{c}{\beta \sin \alpha}$. The length of the first feed-link is bounded by half this length, so any point r has a path of this length or less connecting it to p . So, as long as $|pr| \geq \frac{1}{2}\beta \sin \frac{1}{2}\alpha$, the detour of r is at most $\frac{2c}{\beta^2 \sin \alpha \sin \frac{1}{2}\alpha} < C$, so r will never become a new feed-link point. □

Since D has diameter less than β , each point q_i has an empty (α, β) -triangle t_i with one corner at q_i and the other two corners outside D . Figure 18(a) illustrates the situation, where the grey part lies inside P . Let d_i be the direction of the bisector of t_i at q_i . We prove that two directions d_i and d_j differ by at least $\frac{1}{2}\alpha$.

Lemma B.5. *The angle between d_i and d_j is at least $\frac{1}{2}\alpha$.*

Proof. Assume for contradiction that the angle between d_i and d_j is smaller than $\frac{1}{2}\alpha$. Figure 18(b) shows this situation. First, we will show that the empty (α, β) -triangles t_i and t_j intersect. Consider the empty cones of angle α emanating from q_i and q_j , by extending their



empty triangles. By assumption, their intersection is a cone with an angle larger than $\frac{1}{2}\alpha$ and apex s . Triangle $q_i q_j s$ has base at most $2R$ and top angle at least $\frac{1}{2}\alpha$. Therefore, the side lengths $|q_i s|$ and $|q_j s|$ are at most $\frac{2R}{\sin \frac{1}{2}\alpha} = \beta$. Therefore, s will belong to the (α, β) -triangles belonging to q_i and q_j .

Now, assume that $i < j$, so $|pq_i| < |pq_j|$. We will show that the distance from q_j to p via q_i and its feed-link is short enough to give q_j a good detour, which contradicts the fact that it was later also chosen as a feed-link point.

Consider the path $P[q_i, q_j]$ between q_i and q_j (see Figure 18(c)), the straight line segment $|q_i q_j|$, and the geodesic shortest path $G[q_i, q_j]$ between q_i and q_j inside P . Let the length of the geodesic be denoted $\mu_G(q_i, q_j)$.

Firstly, we know that $\mu_G(q_i, q_j) < \frac{1}{\sin \frac{1}{2}\alpha} |q_i q_j|$, because this path must stay inside triangle $q_i q_j s$. Next, by Lemma B.2, we know that $\mu(q_i, q_j) < \frac{2\pi}{\alpha \sin \frac{1}{4}\alpha} \mu_G(q_i, q_j)$ (note that each edge of the geodesic path is shorter than β and lies P). We conclude that $\mu(q_i, q_j) < g(\alpha) |q_i q_j|$, where $g(\alpha) = \frac{2\pi}{\alpha \sin \frac{1}{4}\alpha \sin \frac{1}{2}\alpha}$. However, the detour factor of q_j via q_i would have been

$$\begin{aligned} \delta_{q_i}(q_j) &= \frac{|pq_i| + \mu(q_i, q_j)}{|pq_j|} \leq \frac{|pq_i| + g(\alpha) |q_i q_j|}{|pq_j|} \\ &\leq \frac{|pq_i| + g(\alpha) (|pq_i| + |pq_j|)}{|pq_j|} \\ &\leq \frac{|pq_j| + g(\alpha) (|pq_j| + |pq_j|)}{|pq_j|} = 1 + 2g(\alpha) < C, \end{aligned}$$

so there was no need for a new feed-link at q_j . \square

Now, we are finally ready to prove the final result.

Theorem 4.5 *Given the boundary P of an (α, β) -covered polygon and a point p inside it, $\frac{4\pi}{\alpha}$ feed-links are sufficient to achieve a dilation of $\frac{4\pi c}{\beta^2 \alpha \sin \alpha \sin \frac{1}{2}\alpha}$, for some absolute constant $c > 0$.*

Proof. We place feed-links incrementally as described, until all points on P have detour at most C . By Lemma B.5 there cannot be more than $\frac{4\pi}{\alpha}$ feed-links, because otherwise some pair q_i and q_j would have (α, β) -triangles with directions d_i and d_j whose angle is smaller than $\frac{1}{2}\alpha$. \square