Computing a Shortest k-Link Path in a Polygon

Joseph S. B. Mitchell * Christine Piatko † Esther M. Arkin ‡

Abstract. *We consider the problem of finding a shortest polygonal path from s to t within a simple polygon P, subject to the restriction that the path have at most k links (edges). We give an algorithm to compute a k-link path with length at most $(1 + \epsilon)$ times the length of a shortest k-link path, for any error tolerance $\epsilon > 0$. Our algorithm runs in time $O(n^3k^3 \log(Nk/\epsilon^{1/k}))$, where N is the largest integer coordinate among the n vertices of P.*

We also study the more general problem of approximating shortest k-link paths in polygons with holes. *In this case, we give an algorithm that returns a path with at most $2k$ links and length at most that of a shortest k-link path; the running time is $O(kE^2)$, where E is the number of edges in the visibility graph.*

Finally, we study the bicriteria path problem in which the two criteria are link length and "total turn" (the integral of $|\Delta\theta|$ along a path). Here, we obtain an exact polynomial-time algorithm for polygons with holes.

1 Introduction

There have been many algorithms in computational geometry that have addressed various notions of shortest paths. The problem of finding shortest (Euclidean length) paths among obstacles in the plane is well-studied [3, 10, 11, 14] and there has also been recent work on the problem of finding shortest paths according to other notions of "length", including L_1 and fixed orientation metrics [5, 12], *link distance* [13, 15], etc.

In this paper, we introduce a new set of results to the field of geometric shortest path algorithms: we study "bicriteria" path problems in which there are *two* objectives in the selection of a path. In particular, we examine the problem of finding a path that is "good" with respect to link distance *and* Euclidean length, and give the first algorithmic results for the problem of finding a shortest k-link path in a polygon, when there are no restrictions on the orientations of links. We also give efficient algorithms for finding a k-link path with minimum "total turn" (defined to be the integral of $|\Delta\theta|$ along the path).

The problems that we address arise due to the non-uniqueness of k-link paths; in fact, there may be an infinite number of different k-link paths joining a given pair of points in a polygon. Even for the case of k equal to the link distance between the pair of points, there is, in general, a continuum of k-link (i.e., minimum-link) paths. This phenomenon is in contrast to the situation with geodesic (Euclidean shortest) paths, where the number of shortest paths joining two points is almost surely equal to one.

One of the main technical issues we must address is the fact that shortest k-link paths (and also minimum-turn k-link paths) need not lie on a "visibility graph", or on any simple variant thereof. Shortest k-link paths may, in general, turn at points interior to the polygon whose coordinates are given by the roots of high-degree polynomials. Thus, we are unable to obtain a path that exactly minimizes the Euclidean length among all k-link paths; instead, our methods produce a *provably good* approximation of a shortest k-link path, whose length is at most $(1 + \epsilon)$ times the length of an optimal path. (Note that, by expressing paths as formulas in the theory of real closed fields, one can obtain an (*exponential time*) exact solution for the decision problem: Does there exist a k-link path of length at most d?) In contrast, in the case of finding a minimum-total-turn k-link path, we obtain an exact optimal solution.

Motivation. In most applications that require one to compute an "optimal" path, there is more than one criterion by which path quality is to be measured. For example, Euclidean shortest paths among obstacles may be too "kinked", with too many links or too much total curvature, for the motion of a mobile robot through an obstacle course. On the other hand,

*Applied Math, SUNY, Stony Brook, NY 11794-3600. Partially supported by NSF grant ECSE-8857642, by Air Force Office of Scientific Research contract AFOSR-91-0328, and by a grant from Hughes Research Laboratories. email: jsbm@ams.sunysb.edu

†Computer Science, Cornell University, Ithaca, NY 14853. Partially supported by NSF Grant ECSE-8857642, Air Force Office of Scientific Research contract AFOSR-91-0328, and an NSF Graduate Fellowship. email: piatko@cs.cornell.edu

‡Applied Math, SUNY, Stony Brook, NY 11794-3600. Partially supported by NSF Grants DMC-8451984 and ECSE-8857642. email: estie@ams.sunysb.edu.

0-8186-2900-2/92 $3.00 © 1992 IEEE

a minimum-link path can be arbitrarily long (in Euclidean length) compared with a shortest length path. (Refer to Figure 2.) Thus, a naturally arising question was posed by Suri: How can one find a shortest (in Euclidean length) path that uses at most k links?

Summary of Results.

1. Using a combination of dynamic programming and a form of binary search (whose "tests" are based on tracing locally optimal shortest k-link paths), we devise an algorithm to compute a k-link path, from s to t, within a simple polygon P (without holes) whose Euclidean length is at most $(1+\epsilon)$ times the length of a shortest k-link s-t path. The algorithm runs in time $O(n^3k^3 \log \frac{Nk}{\epsilon^{1/k}})$, where N is the largest (integer) coordinate among the n vertices of P. (Note that the dependence on ϵ is *logarithmic* in $1/\epsilon$.)

2. For the more general problem of finding short k-link paths in polygons *with* holes, we obtain an algorithm that produces a path with at most $2k$ links and a length at most that of a shortest k-link path; the time complexity is $O(kE^2)$, where E is the number of visibility graph edges. A specialized variant of this algorithm allows us to give an alternative method for simple polygons, where we can find a k-link path (not $2k$-link) with length at most $(1+\epsilon)$ times optimal, in time $O(k^3n^2/\epsilon)$. Compared to the result mentioned above, this method has substantially inferior complexity with respect to its dependence on ϵ; but it gives an improved dependence on n, and it has no dependence on N, the size of the coordinates.

3. We study the problem of finding a k-link path from s to t in a polygon with holes, such that the *total turn* of the path is minimized. The total turn is defined as the integral of $|\Delta\theta|$ along the path, and can be thought of as the total motion of the "steering wheel" when driving a vehicle along the path. We obtain an exact algorithm, whose running time is $O(E^3 \log^2 n)$. Our algorithm is based on a method of finding minimum-link *right-turning* (or left-turning) paths among obstacles in the plane.

Related Work. In [2], we began a study of geometric bicriteria path problems, obtaining several negative results that parallel those known for graphs [7], as well as a pseudo-polynomial time algorithm for the problem of minimizing total turn and Euclidean length.

Recently, de Berg et al. [6] have studied the problem of finding a shortest rectilinear path among rectilinear obstacles, where the measure of path cost is a "combined metric" equal to the sum of the path length (L_1 distance, since they work in a rectilinear world) and a constant times the number of turns; they obtain an $O(n^2)$ algorithm. Hershberger and Snoeyink [9] have examined the special case of finding a shortest k^*-link path (where k^* denotes the number of links in a minimum-link path), for paths and obstacle boundaries required to have edges from among a given fixed set of orientations. Yang, Lee, and Wong [16] study a general class of rectilinear planar problems in which the obstacle set is rectilinear, and path costs are a function of L_1 length *and* the number of bends (rectilinear link distance); they obtain worst-case time bounds of $O(n \log^2 n)$ for the combined metric or for finding a shortest minimum-link path. All of these results on bicriteria paths in a fixed-orientation world exploit the fact that there exists a *path-preserving graph* that is guaranteed to contain the desired optimal path.

Related to our problem of optimizing the length of k-link paths is the problem of computing optimal enclosing k-gons. Aggarwal et al. [1] study the minimum *area* enclosing k-gon problem, obtaining an exact algorithm of time complexity $O(n^2 \log k \log n)$. In his thesis, Chang [4] discusses some of the local optimality properties that minimum *perimeter* enclosing k-gons must satisfy, but leaves open the problem of actually computing them for $k > 3$. It is likely that our methods discussed here will be applicable to this open problem.

Our Method. In contrast to previous work, our problem of finding shortest k-link paths *without orientation restrictions* admits no finite path-preserving graph. Instead, we perform an analysis of the local and global structure of shortest k-link paths in simple polygons. We then use global structure to decompose the problem into a set of "raceway" problems, which we then further decompose into $O(n^2)$ "elementary" problems, which ask us to find shortest i-link paths between pairs of "flush" edges (defined below). We solve these problems by a binary search whose tests consist of tracing locally optimal paths. We combine elementary problem solutions and the raceway solutions into a global solution by means of dynamic programming.

In polygons with holes, we apply a "path preserving graph" approach, constructing a special graph whose embedding has the property that it contains a $2k$-link path whose length is within factor $(1+\epsilon)$ of the length of a shortest k-link path.

2 Preliminaries

Geometric Notations. Let P be a polygon (possibly with holes) in the plane, with a total of n vertices.

We assume that the coordinates of the vertices of P are positive integers; we let N denote the largest x- or y-coordinate among the vertices.

Given two points s and t in P, a *minimum-link path* between them is a polygonal path inside P with a minimum number of edges (or, "links"). The *link distance* $d_L(s,t)$ refers to the number of links in a minimum-link path from s to t.

The visibility graph of P, denoted $VG(P)$, is the graph on the vertex set of P whose edges join mutually visible pairs of vertices. $VG(P)$ can be computed in time $O(E + n \log n)$, where E denotes the number of edges in the graph [8].

It is well-known (e.g., [10]) that, for any two points s and t in a *simple* polygon P, the (Euclidean) shortest path, also called the "geodesic" path, $\pi_G(s,t)$ is unique, lying in $VG(P)$. A minimum-link path, on the other hand, is generally not unique, even in a simple polygon. Indeed, it is precisely the non-uniqueness of k-link paths that makes the problem addressed in this paper interesting.

In a simple polygon P, we define uniquely one particular minimum-link path, which we call the *greedy* path, $\pi_L(s,t)$, from s to t. Specifically, the path $\pi_L(s,t)$ is obtained by using the "windows" (and their extensions) defined by the *window partition* of P with respect to s (as defined in [15]), where the last link is chosen to pass through the last vertex of the geodesic path $\pi_G(s,t)$.

Given a polygonal path π, an interior edge $e \in \pi$ is called an *inflection edge* if the predecessor and the successor edges of e in π lie on opposite sides of the line containing e.

Throughout this paper, the points $s,t \in P$ will remain fixed. We assume that s and t are in general position, so that $\pi_G(s,t)$ is uniquely defined. We also fix the number k to be some integer satisfying $d_L(s,t) \leq k \leq |\pi_G(s,t)|$, where $|\pi_G(s,t)|$ denotes the number of edges in the geodesic path $\pi_G(s,t)$.

3 A General Approximation Scheme

Theorem 3.1 *Let P be a polygon, possibly with holes, whose boundary consists of n edges. For any k, $d_L(s,t) \leq k \leq |\pi_G(s,t)|$, one can compute an s–t path in P that has at most $2k$ links and has length at most that of a shortest k-link s–t path, in time $O(kE^2)$, where E is the number of visibility graph edges.*

Proof Outline. We construct a discrete graph, based on an arrangement of line segments within P, that is guaranteed to contain the claimed $2k$-link path.

For each vertex v of P (and also for $v = s,t$), and for each vertex u that is visible from v, extend a segment out from v in the direction of u, stopping when we enter the exterior of P at some point u'; we say that $\overline{vu'}$ is the segment *anchored at v in the direction of u* and we say that $\overline{uu'}$ is the *extension segment associated with $\overline{vu'}$*. This yields a set S of $O(E)$ segments within P, where E is the size of the visibility graph, $VG(P)$.

We now construct a graph $\mathcal{G}$ whose nodes correspond to the segments S. We connect two nodes with an edge in $\mathcal{G}$ if the associated extension segments cross (at a point interior to P). The length of such an edge, joining nodes corresponding to segments σ_1 and σ_2 (anchored at vertices v_1 and v_2, $v_1 \neq v_2$), is defined to be the length of the path v_1, $\sigma_1 \cap \sigma_2$, v_2. We also connect node σ to node σ' with a zero-length edge, if σ and σ' are anchored at a common vertex v. We show the following claim:

Claim: (a). $\mathcal{G}$ is connected; and (b). corresponding to any shortest k-link s–t path (of length d) in P there is a path in $\mathcal{G}$ with at most $2k$ nodes, such that the corresponding segments constitute a $2k$-link s–t path with length at most d.

$\mathcal{G}$ has $O(E)$ nodes and $O(E^2)$ edges, and it is constructible in time $O(E^2)$. We can then use dynamic programming to compute a shortest $2k$-node path in time $O(kE^2)$. □

Theorem 3.2 *Let P be a simple polygon (without holes) whose boundary consists of n edges. For any k, $d_L(s,t) \leq k \leq |\pi_G(s,t)|$, one can compute, in time $O(k^3n^2/\epsilon)$, an s–t path in P that has at most k links and has length at most $(1+\epsilon)$ times that of a shortest k-link s–t path.*

Proof Outline. We use the fact that the Euclidean metric can be approximated to accuracy ϵ by a fixed-orientation metric of $K = O(1/\sqrt{\epsilon})$ equally-spaced directions. For each of these K directions out of a vertex $v \in \pi_G(s,t)$, we extend a greedy path out for k links in the direction of t. We claim that the resulting arrangement of nkK segments is sufficiently "rich" to contain an approximating k-link s–t path. The proof involves showing that a shortest k-link path can be perturbed onto the arrangement by a sequence of "twists", in which each link is rotated slightly (by at most angle $2\pi/K$) onto a segment of the arrangement, *while maintaining the connectivity of the k links.* □

We will show that in a *simple* polygon P, the dependence on ϵ can be made *logarithmic* in $1/\epsilon$, at a cost of a factor of n and dependence on the term $\log N$. First, we need a careful analysis of the structure of optimal paths.

4 Characterizations

We let π^* denote a shortest k-link path from s to t, and we let d^* denote the (Euclidean) length of π^*. Our goal now is to give a characterization of the structure of π^*; most of the proofs are complex, and are omitted in this extended abstract.

Let the inflection edges of $\pi_G(s,t)$ be denoted $e_1, \ldots, e_I$; let e_0 (e_{I+1}) denote the first (last) link of $\pi_G(s,t)$. The edges e_j partition $\pi_G(s,t)$ into subchains that are purely left-turning or purely right-turning. We call these subchains *convex chains*, although it should be understood that they can spiral, etc. See Figure 3.

Lemma 4.1 *Let π^* be a shortest k-link path in a simple polygon P. Each inflection edge e_j in the geodesic path $\pi_G(s,t)$ must be a subset of an edge of π^*, and each inflection edge of π^* must contain an inflection edge of $\pi_G(s,t)$.*

Remark. The above lemma holds for polygons with holes, provided that we define $\pi_G(s,t)$ to be the shortest path homotopically equivalent to π^*.

For paths within a simple polygon P, we say that an edge in a polygonal s–t path is *flush* if it contains a non-inflection edge of $\pi_G(s,t)$. (A similar definition applies to polygons with holes, provided we use a homotopically equivalent geodesic path.) The following uniqueness result will be a consequence of our various results on local optimality:

Lemma 4.2 *Let a and b be consecutive flush edges along a shortest k-link path, π^*, and let i denote the number of links along π^* between a and b. Then, the subpath of π^* joining a and b is the unique shortest i-link path from a to b.*

Remark. It is not the case that shortest k-link s–t paths in a simple polygon are unique; indeed, in the full paper we give a construction to show that there can be an exponential number of such paths.

An edge of π^* that is not inflection nor flush is called *rocking* if it is in contact with a (single) vertex of P (a *rocking vertex*).

The bend points of path π^* are either *interior* bend points (interior to P) or *boundary* ("bash") bend points (lying on some boundary edge of P). We call an edge of P that contains a bash point a *bash wall* of P.

Let π be a polygonal s–t path and let π' be a subpath of π joining two consecutive interior bend points, p and q, of π. (Thus, the bend points of π', if any, are all bash points.) Let p' be the predecessor of p along π and let q' be the successor of q along π. Assume that all edges of π' are rocking. Refer to Figure 4. We say that π' is a *balanced chain with respect to* π if the path from p' to q' via π' is shortest among paths with the same number of links. In the special case that π' is a balanced chain consisting of the single edge $\overline{pq}$, we say that $\overline{pq}$ is a *balanced edge.*

We can characterize balanced chains as follows: The first link of π' rests on a rocking point r; if we perturb it by a small rotation about r, while adjusting all of the other links of π' accordingly (so that π' remains connected, with all of its bend points being bash points), then the resulting perturbed path from p' to q' must be longer than the original. By somewhat involved calculus, we can quantify this statement and derive a precise characterization of balanced chains:

Lemma 4.3 (Balanced Chains) *In order for a chain π' be to balanced with respect to a polygonal path π, the angle of turn ϕ at its endpoint q must be in a unique relationship with its angle of turn γ at p: $\phi = F(\gamma)$, where the function F depends on p', q', the rocking points of π', and the edges of P containing the bash points of π'. For the case of balanced edges, this relationship is expressed by the relationship*

$$x \cot \frac{\gamma}{2} = y \cot \frac{\phi}{2},$$

where x and y are defined in Figure 5. More generally, the expression

$$\sum_{i=1}^{k-2} \left[\left[y_i \cot \frac{\phi_i}{2} - x_i \cot \frac{\gamma_i}{2} \right] \prod_{j=2}^{i} \frac{h_j \csc^2(\gamma_{j-1} - A_j)}{H_j \csc^2 \gamma_j} \right]$$

must equal zero, where the various constants are defined in Figure 6.

Lemma 4.4 (Local Optimality) *If one perturbs any single edge e of a shortest k-link path, π^*, the result is either to disconnect π^* or to increase its length. In particular, e must be in contact with at least one vertex of P and e must fall into one of the following cases:*

- *e is an inflection edge of π^*, containing an inflection edge of $\pi_G(s,t)$; or*
- *e is a flush edge, containing $\overline{uv} \subset \pi_G(s,t)$, and the balanced position of e with respect to rocking point u is clockwise from e, while the balanced position of e with respect to rocking point v is counterclockwise from e; or*
- *e is a rocking edge with both of its endpoints being bash points; or*

e is a rocking edge with both of its endpoints being interior bends, such that e is balanced; or

e is a rocking edge with exactly one of its endpoints being a bash point — the balanced position of e with respect to its rocking point is either clockwise or counterclockwise from e depending on whether the bash endpoint is on e's left or right end, respectively.

Based on the above lemmas, we can now outline the general structure of an optimal path π^*. Path π^* is partitioned by its inflection edges into pieces, $\pi_0^*, \pi_1^*, \ldots, \pi_{I+1}^*$, with piece π_j^* joining inflection edge e_j to e_{j+1}. Path π_j^* is further partitioned according to the flush edges of π^* into *elementary paths* joining pairs of consecutive flush or inflection edges. Each elementary path joining flush/inflection edge a to flush/inflection edge b has the following structure: From a, π^* is "greedy" (following the bash points of $\pi_L(s,t)$) for some portion, until a *first interior bend point* p_1 (which may, in fact, occur on the first link of the elementary path — i.e., the greedy path may be empty). π^* then consists alternately of chains of interior bend points (i.e., chains of balanced rocking edges) and balanced chains of bash points. See Figure 8 for an example of the structure of an optimal path.

The first interior bend point, p_1, will play a special role in our search algorithm, since we perform a binary search to identify p_1 (approximately) such that the resulting path that can be traced by local optimality hits our desired target in the required number of links. Not all points along $\pi_L(s,t)$ are possible first bend points for π^*. A first bend point $p_1 \in \pi_L(s,t)$ is said to be *valid* if there exists a shortest i-link path from s to some t' that uses p_1 as first bend point. We characterize the valid first bend points as follows: Let $e = \overline{qz}$ be an edge of $\pi_L(s,t)$, with rocking point r. Let p_e be the point on $\overline{rz}$ (if it exists) such that $\overline{qp_e}$ is balanced with respect to the three-link path: predecessor of e, $\overline{qp_e}$, and the ray from p_e through its (left) tangency point, $r_2 \in \pi_G(s,t)$. Refer to Figure 7. One can show

Lemma 4.5 *The valid first bend points for $e \in \pi_L(s,t)$ are precisely the points on the (possibly empty) segment $\overline{p_e z}$.*

If interior turns before p_e were allowed, the path could be locally improved by rocking edge e. By disallowing invalid turns we know that we will be tracing only locally optimal paths.

5 Computing Shortest k-Link Paths in Simple Polygons

Overview of Method. The inflection edges of the (unique) geodesic path $\pi_G(s,t)$ permit us to decompose the full problem into a sequence of "raceway" problems, where we must find (nearly) shortest i-link paths (for many values of i) between two consecutive inflection edges, while staying within a relevant subset of P. Within a raceway, though, there still may be an exponential number of shortest i-link paths; thus, we further decompose the problem into "elementary" problems of finding shortest i-link paths between pairs of flush edges (edges that contain an edge of $\pi_G(s,t)$). Each of these problems is solved by a binary search for the first bend point in an optimal path: each test of the search involves tracing a path forward through the raceway, applying the local optimality conditions of Section 3 at each step (several lemmas are needed to justify this). The justification of the search itself is based on a key "Monotonicity Lemma" which shows that paths traced forwards in this way behave "nicely" with respect to changes in the first bend point. The search stops when every edge of the traced path is known to lie within a very tiny range of angles, of size at most $\delta(\epsilon)$. This property is guaranteed if we make the interval of first bend angles small enough — less than $\delta(\epsilon)/D^k$, where D is a "dilation" factor that measures the maximum amount by which a small interval of directions trapping link i can get larger for link $i+1$. We obtain a bound on the dilation D by bounding the change in the ith turn angle with respect to the change of the first angle (the ith angle is determined by the first angle using the local optimality condition). Our bounds and analysis assume that the vertices of P are on an integer grid of size at most N-by-N; this allows us to write lower bounds on angles between triples of vertices, etc.

Finally, we assemble all of the data from the binary searches, combining it via dynamic programming recurrences to obtain the final result: an ϵ-optimal k-link path from s to t.

5.1 Decomposing the Problem

Using Lemma 4.1, we can decompose our problem into a sequence of "raceways" between extensions of inflection edges. Raceway R_j is the portion of P that lies between inflection edges e_j and e_{j+1}. Refer to Figure 9. We further trim from R_j all of the pockets of P that are on the convex side of the subpath $\pi_G(s_j, t_j)$. (A simple lemma justifies that π^* will never enter such pockets.)

We can also replace the portion of the boundary of P between s_j and t_j (going clockwise) by the *complete visibility chain*, C_j, which is the boundary of all points p such that p sees both of its tangent points with π_j. (C_j can be found in linear time.) We show that π^* must lie within the raceway R_j, which is bounded by C_j, the inflection edges e_j and e_{j+1}, and the path π_j.

Thus, we can decompose our problem into finding shortest k_j-link paths between s_j and t_j in a raceway R_j, where k_j ranges from $d_L(s_j, t_j)$ up to $|\pi_G(s_j, t_j)|$.

5.2 Shortest k-link Path in a Raceway

Let us fix our attention on one raceway, R. We desire shortest i-link paths within R, for a variety of values i (at most k such values). By results of Section 3, a shortest i-link path may have many flush edges along it, but between each pair of consecutive flush edges, the path is an alternating sequence of chains of bash points (greedy paths) and interior-bend paths.

We introduce $O(n)$ additional special edges that further decompose the raceway problem. We call these edges *leaning* edges, and they are defined to be edges $\overline{uv}$ of $VG(P)$ such that one endpoint u is a vertex of $\pi_G(s,t)$, while v is a vertex of P such that $\overline{uv}$ is an edge of *right tangency* from v, with respect to the chain $\pi_G(s,t)$ within R. Effectively, the leaning edges can "split" our "wedge" of traced paths, causing the angular ranges of each link to amplify in an unbounded fashion (preventing us from obtaining a bounded "dilation" factor). Thus, we further subdivide the raceway problem according to the $O(n)$ leaning edges, and we are able to show that the resulting elementary problems from a to b (where a and b are now either inflection, flush, or leaning edges) permit a provably good approximation to the optimal path.

We define the *combinatorial type* of a path to be the sequence of labels, one per bend point, listing the bash wall for each bashing bend point, or noting that a bend point is interior. (Note that we do *not* include the set of rocking points in this definition of combinatorial type.) Leaning edges correspond, then, to edges of $VG(P)$ that can account for a change in the combinatorial type of our traced paths.

5.3 The Main Search Algorithm

Let a and b each be a flush, leaning, or inflection edge, where b follows a along $\pi_G(s,t)$. Let s_a be the first vertex of a on $\pi_G(s,t)$ and r_a the succeeding one. Let r_b be the first vertex of b on $\pi_G(s,t)$ and t_b the succeeding one. We now describe the main algorithm *Path*(a,b,i) to find an ϵ-optimal approximation to the shortest i-link path from a and b that does not use an intermediate flush edge.

We begin with the subprocedure that traces paths according to the local optimality conditions.

Tracing Paths. By the local optimality criteria for balanced chains, once we know a *valid* first bend point, p_1, when going from flush edge a to flush edge b, we can trace out uniquely a shortest path consistent with that first bend point. We let $\pi(\gamma, i)$ denote the resulting path, where γ is the turn angle associated with point p_1.

The crucial lemma in being able to do tracing is the following, whose very technical proof is omitted here:

Lemma 5.1 (Tracing Lemma) *Given a first bend point p_1, the next link of $\pi(\gamma, i)$ is uniquely determined (it is simply obtained by "leaning" an edge through p_1 against path $\pi_G(s,t)$). Given a subpath of $\pi(\gamma, i)$, traced some number of links beyond p_1, the next link of $\pi(\gamma, i)$ is* uniquely *determined by the local optimality conditions and the subpath so far. The next rocking point can therefore be determined by advancing along $\pi_G(s,t)$, one vertex at a time, starting from the last rocking point of the current subpath.*

Remark. We advance along $\pi_G(s,t)$ to find the next rocking point, *assuming that the next bend point is an interior turn.* If, once we discover the unique next "interior" bend point, we find that it is *not* reachable while staying interior to P, then the traced path should have a bash point on the end of the current link, so we "bash" and continue. At any stage, we may have a long sequence of bash points as the last several turns in the current subpath. Preceding the sequence of bash points is an interior bend point (e.g., the first such point p_1); thus, we are able to use the balanced-chain condition to determine where the next interior bend should be (if it is feasible), etc.

The Binary Search. Let Γ_0 be the set $\bigcup_{e \in \pi_L} \Gamma_e$, where Γ_e is the subinterval $[\gamma_e, \gamma'_e]$ of valid angles γ that correspond to valid first interior bend points $p_1 \in e$. Let $\pi(\gamma, i)$ for $\gamma \in \Gamma_0$ be the resulting traced path that goes at most i links, or until the first rocking point that is beyond r_b on $\pi_G(s,t)$.

A point p_1 of Γ_0 is identified with the angle γ between $\pi_L(a,b)$ and the ray from p_1 that is tangent to $\pi_G(s,t)$. Note that the total length of Γ_0, $|\Gamma_0|$, is bounded by $\sum_{e \in \pi_L} |\Gamma_e| = \sum_e (\gamma'_e - \gamma_e) \le \pi \cdot d_L(a,b)$.

During this procedure we tabulate $g_i(a,b)$, an ϵ-optimal approximation to the shortest i-link elementary path from a and b (where a and b are flush or leaning edges in the same raceway).

Path(a, b, i):

0. $\Gamma \leftarrow \Gamma_0$

1. If $|\Gamma| \le \delta_0 = \delta(\epsilon)/D^k$, go to Step 3.
   Otherwise, $\gamma \leftarrow$ midpoint of Γ and $\Pi \leftarrow \pi(\gamma, i)$.
   Let $r =$ ith rocking point of Π.

2. If Π has less than i links (i.e. we overshot b), then $\Gamma \leftarrow$ left half of Γ and go to Step 1.
   Otherwise, if Π has i links, and $r = r_b$, then if p is to the left of $\overline{t_b r_b}$
   then $\Gamma \leftarrow$ right half of Γ and go to Step 1,
   else if p is to the right of $\overline{t_b r_b}$ (which can only happen if b is a leaning edge)
   then $\Gamma \leftarrow$ left half of Γ and go to Step 1.
   Otherwise, if r precedes r_b along $\pi_L(a, b)$, $\Gamma \leftarrow$ right half of Γ and go to Step 1.

3. If the combinatorial type of $\pi(\gamma, i)$ for $\gamma \in \Gamma = [\gamma_{\min}, \gamma_{\max}]$ is constant, then $g_i(a, b)$ is assigned the length of $\pi(\gamma_{\max}, i)$.
   Otherwise $g_i(a, b) \leftarrow \infty$, since the ϵ-optimal path from a to b that we will produce will pass through some leaning edge c, and so will be found as $g_{i'}(a, c) + g_{i-i'-1}(c, b)$ for some i'.

Correctness of the Search. Correctness of the above binary search procedure is based on the following key lemma, which is used to justify discarding the relevant half of the interval Γ in Step 2:

Lemma 5.2 (Monotonicity) *Consider a particular raceway problem. Let C denote the corresponding complete visibility profile, and let Γ_0 denote the set of valid first turn angles. Let $z(\gamma)$ denote the point on C obtained by extending the ith link of $\pi(\gamma, i)$ until it exits the raceway. Then, if $\gamma, \gamma' \in \Gamma_0$, with $\gamma < \gamma'$, then $z(\gamma)$ precedes $z(\gamma')$ along the chain C.*

Proof Outline. The proof is by contradiction, and is based on the fact that $\pi(\gamma, i)$ gives an (exact, unique) shortest k-link path from a to any point of $\pi(\gamma, i)$ that lies on the kth link of $\pi(\gamma, i)$, *after* the kth rocking point of the path. We consider the various cases of how the two paths, $\pi(\gamma, i)$ and $\pi(\gamma', i)$, can interleave, and obtain contradictions based on simple geometric inequalities and the preceding fact. □

Bounding the Path Length Error. We now sketch the analysis of how we bound the size $|\Gamma|$ in order to guarantee a path quality factor of $(1 + \epsilon)$.

Lemma 5.3 (Trapping Lemma) *For a given a and b, and for a given k, if $\Gamma = [\gamma_{\min}, \gamma_{\max}]$ is such that the paths $\pi(\gamma, k)$ all have common combinatorial type, for $\gamma \in \Gamma$, and the angles between the ith edge of $\pi(\gamma_1, k)$ and the ith edge of $\pi(\gamma_2, k)$ are each bounded above by δ, then the Euclidean length of $\pi(\gamma, k)$ is at most $(1 + O(\delta N^2))$ times the length of a shortest k-link path joining a and b.*

Corollary 5.4 *For ϵ-optimality, it suffices to select $\delta(\epsilon) = O(\frac{\epsilon}{N^2})$.*

Based on careful analysis of the balanced-chain condition, we are able to bound the dilation factor:

Lemma 5.5 (Dilation) *Let θ_j denote the orientation of the jth link of path $\pi(\gamma, i)$, and let $\theta_j = F(\gamma)$ denote the functional dependence on the first interior bend angle, γ. Then, $|F'(\gamma)| \le k^{O(1)} N^{O(k)}$, implying that the dilation $D = k^{O(1)} N^{O(k)}$.*

Lemma 5.6 *All of the values $g_i(a, b)$ can be computed in total time $O(n^3 k^3 \log(Nk/\epsilon^{1/k}))$, where N is the largest integer coordinate of any vertex of P.*

Proof Outline. There are $O(n^2 k)$ values of $g_i(a, b)$ to tabulate. Each requires a binary search with

$$\begin{aligned} O\left(\frac{|\Gamma_0|}{\delta_0}\right) &= O\left(k \log \left(\frac{k}{\delta(\epsilon)}\right)^{1/k} D\right) \\ &= O\left(k^2 \log \frac{Nk}{\epsilon^{1/k}}\right) \end{aligned}$$

tests, with each test requiring time $O(n)$. □

5.4 Putting the Pieces Together

We now show how the path pieces computed so far can be assembled into a global solution by means of dynamic programming. We have tabulated $g_i(a, b)$, an ϵ-optimal approximation to the shortest i-link elementary path from a and b (where a and b are flush or leaning edges in the same raceway) for all values of i.

We now find an ϵ-optimal approximation to a shortest k-link path from s to t in two stages. In the first stage we compute $f_i(j)$, the length of the ϵ-optimal approximation to the shortest i-link path in raceway j. Using these values, we compute $h_i(j)$, the ϵ-approximation to the shortest i-link path from e_j to t, where e_j is an inflection edge. This gives us an ϵ-optimal path from s to t.

For the first stage, let us fix the raceway R_j, bounded by inflection edges e_j and e_{j+1}. Define $F(a, i)$ to be the ϵ-optimal length of the i-link path from a to e_{j+1}, where a is a flush or leaning edge in raceway j. We need to tabulate a total of $O(nk)$

such values. Initialize $F(a,i)$ to be $g_i(a, e_{j+1})$. Compute $F(a,i) = \min\{\min_{m,c}\{g_m(a,c) + F(c, i-m+1)\}, F(a,i)\}$ for $a < c < e_{j+1}$ (and for c which are leaning edges with rocking points between a and e_{j+1}) and $1 < m < i$, with a advancing backwards along $\pi_G(s,t)$ from e_{j+1} to e_j. There are $O(kn)$ choices for m and c, so it takes total time $O(n^2k^2)$ to compute all the values $f_i(j) = F(e_j, i)$. Summing over all $O(k)$ possible raceways, it takes time $O(n^2k^3)$ to compute all values $f_i(j)$.

For the second stage let $h_i(j)$ be the ϵ-approximate length of a shortest i-link path from e_j to t. We can compute these recursively by using the following: $h_i(j) = \min_{i'}\{f_{i'}(j) + h_{i-i'}(j+1)\}$. Thus, we can tabulate all the $h_i(j)$'s in $O(k^3)$ time.

The time bounds for these two stages are dominated by the time to tabulate the values $g_i(a,b)$ (the elementary paths between flush edges and leaning edges).

Theorem 5.7 *One can compute an ϵ-optimal shortest k-link path in an n-sided simple polygon P in time $O(n^3k^3\log(Nk/\epsilon^{1/k}))$, where N is the largest integer coordinate of any vertex of P.*

6 Minimum Total-Turn k-Link Paths

Consider a polygon P with holes. One measure of how "kinked" a path is can be based on the "total turn" (TT), defined to be the integral of $|\Delta\theta|$ along the path, where θ denotes the orientation of the tangent vector. The TT measures the total integrated motion of the steering wheel if we drive a car along the path, always making the turn at a vertex according to the angle that is less than 180 degrees. In [2], we showed that the bicriteria path problem of minimizing (TT, Euclidean length) is (weakly) NP-hard; we also provided a pseudo-polynomial time algorithm.

We provide an exact polynomial-time algorithm for the bicriteria path problem with objectives (TT, link distance), with running time $O(E^3\log^2 n)$.

While we have no room here to give details, we sketch our method as follows. We identify all inflection edges of the visibility graph (i.e., VG edges that are locally supporting on opposite sides of the edge). We prove a structural result that a path π^* that is optimal for our two criteria must contain the inflection edges that lie along the geodesic path homotopically equivalent to π^*. Between two inflection edges, then, the path π^* should be purely left-turning or purely right-turning. The total turn along the various purely left-turning paths that may join an inflection edge a to another inflection edge b are all equal to the difference in angles of a and b, $|\theta_a - \theta_b|$, plus $360i$ degrees, for $i = 1, 2, 3, ..., I$ (where $I \leq n$). Note that there may in fact be an exponential number of homotopically distinct, locally optimal, left-turning paths from a to b. Our goal, then, is to compute a minimum-link left-turning path from a to b, for each value of i, and for each of the $O(E^2)$ pairs a and b of inflection edges. We accomplish this by a modification of the algorithm of Mitchell, Rote, and Woeginger [13], in which we perform *constrained illumination* at each step, allowing only those paths that are purely left-turning (or purely right-turning). The algorithm of [13] runs in time $O(E\log^2 n)$, as does our modification. This gives a total time of $O(E^3\log^2 n)$ for computing the connections between inflection edges. We then combine these TT costs/link lengths via a dynamic programming recursion (time $O(kE^2)$) to obtain the final result.

References

[1] A. Aggarwal, J. S. Chang, and C. K. Yap. Minimum area circumscribing polygons. *Visual Comput.*, 1:112–117, 1985.

[2] E. M. Arkin, J. S. B. Mitchell, and C. D. Piatko. Bicriteria shortest path problems in the plane. In *Proc. 3rd Canad. Conf. on Computational Geometry*, pages 153–156, 1991.

[3] Ta. Asano, Te. Asano, L. J. Guibas, J. Hershberger, and H. Imai. Visibility of disjoint polygons. *Algorithmica*, 1:49–63, 1986.

[4] J. S. Chang. Polygon optimization problems. Report 240, CS, New York Univ., New York, NY, 1986.

[5] K. L. Clarkson, S. Kapoor, and P. M. Vaidya. Rectilinear shortest paths through polygonal obstacles in $O(n(\log n)^2)$ time. In *Proc. 3rd Annu. ACM Sympos. Comput. Geom.*, pages 251–257, 1987.

[6] M. de Berg, M. van Kreveld, B. J. Nilsson, and M. H. Overmars. Finding shortest paths in the presence of orthogonal obstacles using a combined L^1 and link metric. In *Proc. 2nd SWAT*, volume 447 of *LNCS*, pages 213–224. Springer-Verlag, 1990.

[7] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, New York, NY, 1979.

[8] S. K. Ghosh and D. M. Mount. An output-sensitive algorithm for computing visibility graphs. *SIAM J. Comput.*, 20:888–910, 1991.

[9] J. Hershberger and J. Snoeyink. Computing minimum length paths of a given homotopy class. In *Proc. 2nd WADS*, volume 519 of *LNCS*, pages 331–342. Springer-Verlag, 1991.

[10] D. T. Lee and F. P. Preparata. Euclidean shortest paths in the presence of rectilinear barriers. *Networks*, 14:393–410, 1984.

[11] J. S. B. Mitchell. A new algorithm for shortest paths among obstacles in the plane. *Annals of Math. and Artificial Intelligence*, 3:83–106, 1991.

[12] J. S. B. Mitchell. L_1 shortest paths among polygonal obstacles in the plane. *Algorithmica*, 8:55–88, 1992.

[13] J. S. B. Mitchell, G. Rote, and G. Woeginger. Minimum-link paths among obstacles in the plane. *Algorithmica*, 8, 1992.

[14] M. Sharir and A. Schorr. On shortest paths in polyhedral spaces. *SIAM J. Comput.*, 15:193–215, 1986.

[15] S. Suri. On some link distance problems in a simple polygon. *IEEE Transactions on Robotics and Automation*, 6(1):108–113, 1990.

[16] C. Yang, D. Lee, and C. Wong. Rectilinear paths among rectilinear obstacles revisited. Technical report, Dept. of EE & CS, Northwestern Univ., 1992.

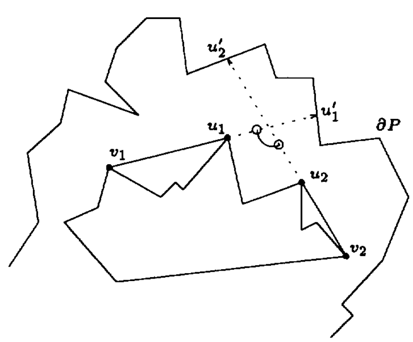

Figure 1: Definition of graph $\mathcal{G}$.

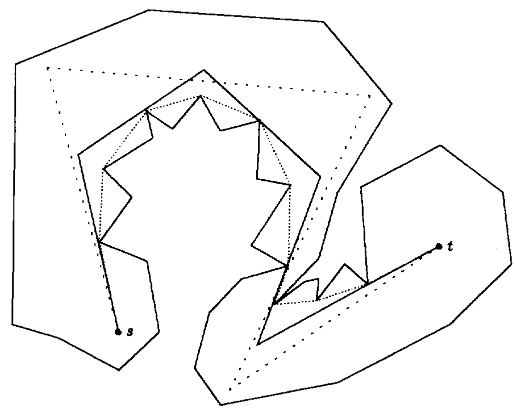

Figure 2: Example of minimum-link path, shortest path, and shortest 5-link path.

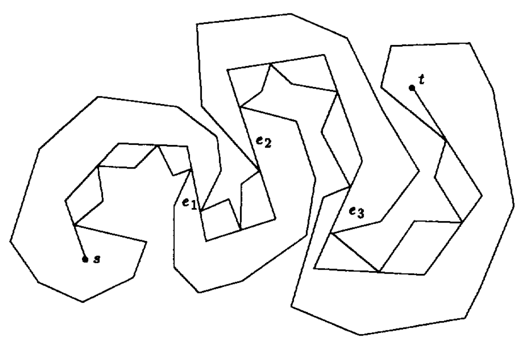

Figure 3: The inflection edges of $\pi_G(s,t)$.

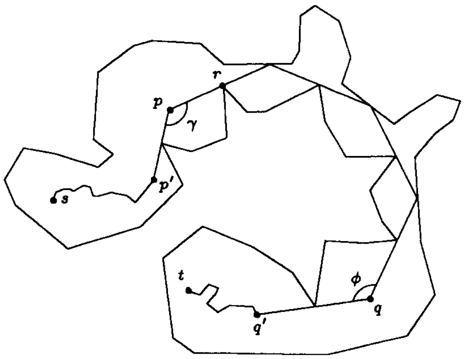

Figure 4: A balanced chain π'.

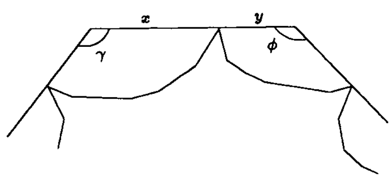

Figure 5: A balanced edge.

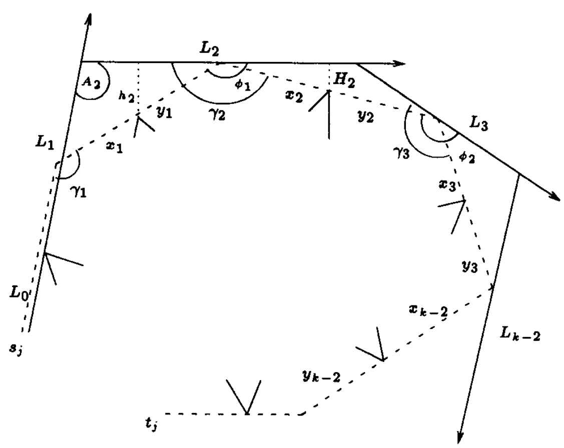

Figure 6: Balanced edges with bashes.

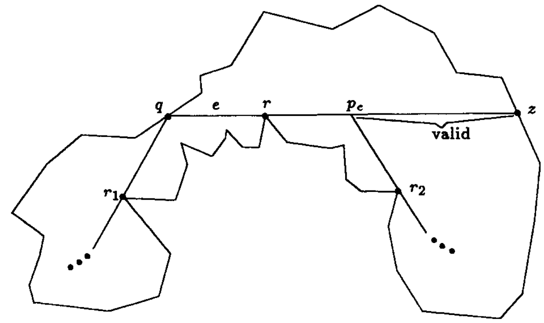

Figure 7: Valid first bend points along edge e.

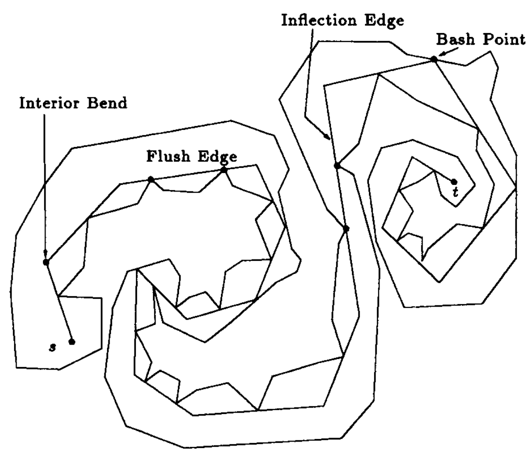

Figure 8: Structure of an optimal path.

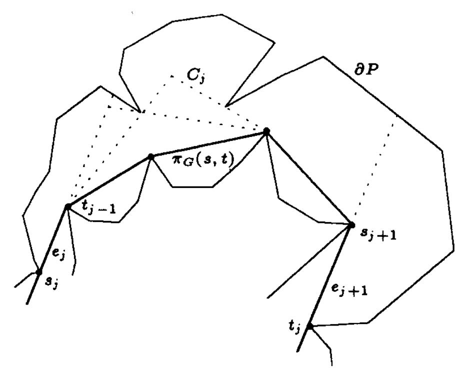

Figure 9: The raceway R_j between inflection edges e_j and e_{j+1}.