Widest Empty Corridor with Multiple Links and Right-angle Turns
(Extended Abstract)

Siu-Wing Cheng[1]
Department of Computer Science
The Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong
scheng@cs.ust.hk

Abstract. We formulate the problem of computing the *widest empty corridor with at most ℓ links and right-angle turns* for a set of n points. It is a generalization of the widest empty corridor problem studied in [2, 3] and it relates to the problem of computing corridors of other shapes posed in [1, 3] (e.g., L-shaped corridor). When $\ell = 2$, it becomes the widest empty L-shaped corridor problem for which we develop an $O(n^3)$-time algorithm. For general ℓ, we present a dynamic programming algorithm and prove a bound of $O(\ell n^8)$ on its running time. We also develop a faster approximation algorithm that computes in $O((1/\epsilon)\ell n^3 \log^5 n)$ time a solution with width at least $(1-\epsilon)$ times the optimal for any $\epsilon > 0$.

1 Introduction

A *corridor* through a point set S is the open region of the plane that is bounded by two parallel straight lines intersecting the convex hull of S. A corridor is empty if and only if it does not contain any point of S [1, 2, 3]. The problem of computing the widest empty corridor through a set of n points have been studied before and it can be done in $O(n^2)$ time [2, 3]. The motivation is to find a collision-free straight route to transport objects through a set of point obstacles. However, even the widest empty corridor may not be wide enough sometimes. This motivates us to consider allowing turns (restricted to right-angle turns in this paper) and generalizing to corridors with *multiple links*. This relates to the problem of computing corridors of other shapes (e.g., L-shaped corridor) posed in [1, 3].

We call an open polygonal region a *regular corridor with ℓ links* if it is the concatenation of ℓ links (to be defined below) such that neighbouring links are perpendicular to each other. The links of a regular corridor may have different widths and the minimum is taken as the width of the corridor. To avoid some trivial solution to our problem, we also require the corridor to satisfy two criteria which will be elaborated in Section 2.

We define a *link* to be an open polygonal region which is bounded by : (1) two parallel lines (type 1 link), or (2) two parallel line segments and two other line segments forming a trapezoid (type 2 link), or (3) two parallel rays and one line segment forming an unbounded trapezoid (type 3 link). See Figure 1 for examples. The *width* of a link is the perpendicular distance between its bounding parallel lines or line segments or rays. Note that although we require neighbouring links to be perpendicular to each other, a link need not be parallel to either of the two axes (i.e., not necessarily orthogonal).

The largest width achievable is a monotonic function in the maximum number of turns allowed. Thus, we are interested in computing the widest empty regular corridor when an upper bound on the number of links is specified. This problem may find application in transporting objects under the circumstance that straight-line movement is fast whereas it is relatively slow to change direction. It is then desirable to select the empty regular corridor with the fewest number of links among the wide enough ones.

In Section 3, we present an $O(n^3)$-time algorithm to compute the widest empty L-shaped corridor. It is based on the observation that it suffices to consider empty L-shaped corridor that can be completely specified by three points and a rotational deformation that can be computed in $O(1)$ amortized time. In Section 4.1, we present an $O(\ell n^8)$-time algorithm to compute the widest empty regular corridor with at most ℓ links. In Section 4.2, we show that the running time can be reduced significantly

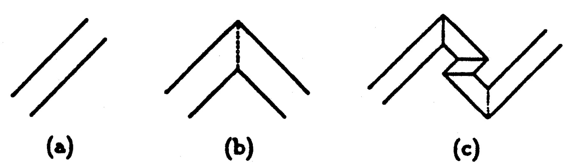
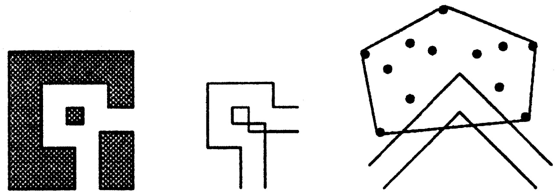


Figure 1. (a) One type-1 link; (b) Two type-3 links; (c) Two type-3 links and three type-2 links

[1]Research supported by the RGC Grant HKUST 190/93.



(a) (b)

Figure 2.

to $O(\ell n^2 \log^5 n)$ if the corridor is restricted to be orthogonal. We then show how to apply this special-case algorithm obtain an $O((1/\epsilon)\ell n^3 \log^5 n)$-time approximation algorithm to compute an empty regular corridor with at most ℓ links and width at least $(1-\epsilon)$ times the optimal for any $\epsilon > 0$.

2 Preliminaries

A regular corridor is allowed to be self-intersecting because there are circumstances in which the widest empty regular corridor will have to be non-simple. See Figure 2(a): The shaded regions are dense with points and the empty corridor on the right is the widest possible. To make sure that the concatenation of links is well defined, we require that if we traverse the two bounding polygonal lines of the corridor simultaneously, then the bounding parallel line segments/rays of each link must be traversed in the same direction. We also require the corridor to satisfy the following two conditions.

Condition C1. Each unbounded exterior region induced must contain some point of S.

Condition C2. P induces two unbounded exterior regions.

The motivation for the conditions is as follows. Unlike the empty 1-link corridor, we cannot simply require the corridor to intersect the convex hull of S because that will allow it to "scratch the exterior" of S without actually passing through S. See Figure 2(b). In this case, the corridor can be arbitrarily wide. Condition C1 prevents the "scratching of the exterior" of S. Condition C2 disallows the first and last links (the two extreme type-3 links) of the corridor to overlap each other when they have opposite directions. That is, the corridor cannot go around a "loop". Condition C2 allows us to prove Lemma 2 in Section 4 which in turns shows the correctness of our algorithm in Section 4. At this point, we do not know if Lemma 2 will hold without condition C2. Nevertheless, we believe that condition C2 is not too restrictive since a corridor intuitively means a passage *through* a region.

3 Widest empty L-shaped corridor

We shall assume that neither of the links of the widest empty L-shaped corridor can be extended to become an empty 1-link corridor. Otherwise, we can transform the widest empty 1-link corridor into a widest empty L-shaped corridor simply by adding an arbitrarily wide link outside the convex hull of S. We call the corridor boundary that contains a convex (resp. concave) corner with respect to the interior of the corridor an *outer* (resp. *inner*) boundary. Each boundary consists of two rays which we call *boundary legs*. We call an empty L-shaped corridor *non-expansive* if each boundary leg contains a point of S and each outer boundary leg contains a point of S in its relative interior.

Lemma 1 *For any empty L-shaped corridor C, there exists a non-expansive empty L-shaped corridor no narrower than C.*

Proof (Sketch) If r is a boundary leg and there is no point of S lying on it (in its relative interior if r is an outer boundary leg), then we claim that we can slide r until r hits a point of S (in its relative interior if r is an outer boundary leg). Otherwise, either the link not bounded by r can be extended to become an empty corridor or there is no point of S lying on one side of C, a contradiction. Hence, one can repeat the above until C becomes non-expansive. □

We first present algorithm $\mathcal{D}$ to construct certain empty L-shaped corridors and then show how to use it to find the widest empty L-shaped corridor.

3.1 Algorithm $\mathcal{D}$

Let S be $\{p_i : 1 \le i \le n\}$. Add the point p_∞ at $y = +\infty$ to S. Then $S \cup \{p_\infty\}$ induces $n(n-1)+n$ vectors from one point in S to a point in $S \cup \{p_\infty\}$. Note that $\overrightarrow{p_2 p_\infty}$[2] is a ray shooting vertically upward from p_2. Imagine that we sweep the positive y-axis for a full circle in the clockwise direction. The direction of each vector will become consistent with the rotating positive y-axis at some time and we sort the vectors in $O(n^2 \log n)$ time according to the chronological order of these events. We examine these vectors in this order. For each vector $\overrightarrow{p_2 p_1}$, we rotate the plane (in the counter-clockwise direction) until $\overrightarrow{p_2 p_1}$ points vertically upward. Let the x-coordinate of $\overrightarrow{p_2 p_1}$ be x_0. We assume for the time being that the following queries can be answered in $O(1)$ amortized time

[2] $\overrightarrow{xy}$ denotes the vector from x to y.

each. We shall explain in Section 3.2 how to support these queries.

1. For any p_i, report the two points that will be hit first if a horizontal (resp. vertical) ray shooting rightward (resp. upward) from p_i is rotated clockwisely and counter-clockwisely.

2. For any p_i, report p_j such that $x(p_j) > x_0$, $y(p_j) > y(p_i)$, and $x(p_j)$ is minimized ($y(p_j)$ is minimized).

3. For any p_i, report p_j such that $x(p_j) < x_0$, $y(p_j) > y(p_i)$, and $x(p_j)$ is maximized.

4. Define $circle(p_i, p_j)$ to be the circle with $\overline{p_i p_j}$[3] as diameter. Given p_i and p_j such that $x(p_i) \geq x(p_j)$ and $y(p_i) \leq y(p_j)$, report the two points inside or on $circle(p_i, p_j)$ that will be hit first if a horizontal (resp. vertical) ray shooting leftward (resp. downward) from p_i is rotated clockwisely and counter-clockwisely.

We construct four types of candidates as follows.

Candidate 1. This candidate handles the case when p_1 and p_2 lie on the outer boundary. See Figure 3(a). For each p_k such that $y(p_k) \leq y(p_2)$ and $x(p_k) \geq x_0$, set the outer boundary to be the L passing through p_1, p_2, and p_k. Use query 3 to find the two points with the minimum x-coordinate and minimum y-coordinate, respectively, in the open convex region enclosed by the outer boundary. Set the inner boundary to be the L passing through the two points found. This produces candidate 1 in $O(1)$ time per p_k.

Candidate 2. From each candidate 1, we construct a candidate 2 by applying a rotational deformation. We describe below the case where $p_1 \neq p_\infty$. The case where $p_1 = p_\infty$ can be handled similarly. Let p_k be the point lying on the horizontal outer boundary leg. Let p_i and p_j be the two points lying on the vertical and horizontal inner boundary legs, respectively. See Figure 3(b). We first rotate the corridor in the clockwise direction while requiring its outer boundary to pass through p_2 and p_k and its inner boundary to pass through p_i and p_j at all times. The rotation terminates as soon as a boundary leg hits a point of S. The amount of rotation allowed can be determined as follows. Use query 1 (resp. query 2) to find the minimum degree θ_1 (resp. θ_2) that the parts of the horizontal (resp. vertical) boundary legs from p_k and p_j to $x = +\infty$ (resp. from p_2 and p_i to $y = +\infty$) can rotate before hitting a point of S. The degree θ_3 (resp. θ_4) that the corner of the outer boundary between p_2 and p_k (resp. the corner of the inner boundary between p_i and p_j) can rotate before hitting a point of S can be computed using queries 5 and 6. The amount of rotation allowed is thus $\min\{\theta_1, \theta_2, \theta_3, \theta_4\}$. Since the width of each link of candidate 1 changes sinusoidally with the degree of rotation, we can determine in $O(1)$ time the widest empty L-shaped corridor C_1 that can be obtained during the clockwise rotation. Similarly, we rotate candidate 1 in the counter-clockwise direction (while requiring its outer boundary to pass through p_1 and p_k and its inner boundary to pass through p_i and p_j at all times) and determine the widest empty L-shaped corridor C_2 that can be obtained. We pick the wider one between C_1 and C_2 as candidate 2.

Candidate 3. This candidate handles the case when p_1 and p_2 lie on the inner boundary. See Figure 3(c). For each p_k such that $y(p_k) < y(p_2)$, use query 4 to find the point p_j that has the maximum x-coordinate while $x(p_j) < x_0$ and $y(p_j) > y(p_k)$. If $x(p_j) \leq x(p_k)$, then set the outer boundary to be the L passing through p_j and p_k. Otherwise, we abort (because p_k does not lie on the outer boundary). If we have not aborted, then use query 3 to find the point with the minimum y-coordinate in the open convex region enclosed by the outer boundary. Set the inner boundary to be the L passing p_1, p_2, and the point found. This produces candidate 3 in $O(1)$ time per p_k.

Candidate 4. This candidate is obtained by applying a rotational deformation to candidate 3. It is similar to computing candidate 2.

Theorem.1 *A widest empty L-shaped corridor for a set of n points can be computed in $O(n^3)$ time.*

Proof (Sketch) Given a widest empty non-expansive L-shaped corridor C, we perform clockwise rotational deformation (as described in algorithm $\mathcal{D}$) until a boundary leg contains two points of S. This possibly deformed C must be a candidate 1 or 3 computed by algorithm $\mathcal{D}$ on S or the mirror image of S obtained by flipping S around the y-axis. Thus, the original C must be a candidate 1 or 2 or 3 or 4. (Some degenerate cases are not covered in this extended abstract.) □

3.2 Preprocessing

For clarity, we use $S_{\overrightarrow{p_2p_1}}$ to denote the set of points after rotating to make $\overrightarrow{p_2p_1}$ pointing upward. S denotes the original set of points.

Queries 1 and 2. For each point p_k in S, we store the other points of S in a doubly linked circular list $list(p_k)$ sorted in angular order around p_k. The total time needed for computing all $list(\cdot)$'s is $O(n^2 \log n)$. Since algorithm $\mathcal{D}$ works on the angular ordering of the vectors induced

[3] $\overline{xy}$ denotes the line segment joining x and y.

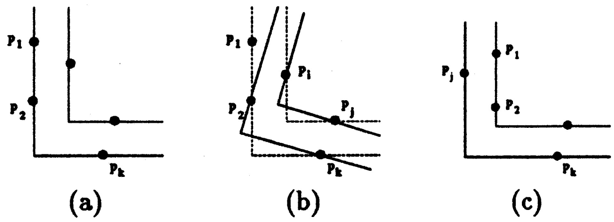


Figure 3.

by $S \cup \{p_\infty\}$, the answers to successive invocations of query 1 to p_k can be answered by traversing *list*(p_k). Hence query 1 takes $O(1)$ amortized time. Query 2 can be similarly supported.

Queries 3 and 4. If $S_{\overrightarrow{p_2p_1}}$ is already sorted in non-increasing y-coordinates, then we can scan $S_{\overrightarrow{p_2p_1}}$ in this order and for each p_k encountered, we can compute and record the answers of applying queries 3 and 4 to p_k in $O(1)$ time. However, we cannot run a sorting algorithm on each $S_{\overrightarrow{p_2p_1}}$ because the total time will then be $O(n^3 \log n)$. S induces $n(n-1)/2$ line segments joining two points of S. First, we obtain a list, *segment*(S), of these line segments sorted by slopes in $O(n^2 \log n)$ time. Third, imagine that we rotate the plane in the counter-clockwise direction and stop when some line segment $\overline{p_ip_j}$ becomes horizontal. This stop can be detected easily since $\overline{p_ip_j}$ must be the first element in *segment*(S). The ordering of points (by y-coordinates) remains the same before this stop and only p_i and p_j should be swapped afterward. We can then continue the rotation to the second stop (second element in *segment*(S)) and so on. If we use a persistent search tree to store the ordering of points, then each swapping takes $O(1)$ amortized time (no rebalancing is needed). Thus, the total time taken is $O(n^3)$.

Queries 5 and 6. For every two points p_i and p_j of S we compute a doubly linked circular list, *cir_list*(p_i, p_j) (resp. *cir_list*(p_j, p_i)) of points lying in *circle*(p_i, p_j) sorted in angular order around p_i (resp. p_j). This can be done by scanning *list*(p_i) (resp. *list*(p_j)) and include a point whenever it is contained in *circle*(p_i, p_j). Since algorithm $\mathcal{D}$ works on the angular ordering of the vectors induced by $S \cup \{p_\infty\}$, answering the invocations of queries 5 and 6 to p_i and p_j correspond to traversing *cir_list*(p_i, p_j) and *cir_list*(p_j, p_i) as in answering query 1. Thus, queries 5 and 6 also take $O(1)$ amortized time.

4 Widest empty regular corridor

We say that a regular corridor is *supported* if each bounding line segment/ray of every link contains a point of S. If the two bounding line segments/rays of a link contains points p_i and p_j respectively, then we say that the link is supported by p_i and p_j. The notion of support is similar to but less stringent than the non-expansive property. A result analogous to Lemma 1 holds for empty regular corridors.

Lemma 2 *For any empty regular corridor, there exists a supported empty regular corridor with at least the same width and no more links.* □

The proof strategy is as follows. Let N_1 be the set of empty regular corridors that have at least the same width as the given empty regular corridor. Pick $N_2 \subseteq N_1$ that minimizes the number of links. Pick $N_3 \subseteq N_2$ that maximizes the sum of widths of links. Then it can be shown that each element of N_3 is supported; otherwise, either condition C1 or condition C2 or the minimality of number of links is violated. The detail involves case analysis and is omitted in this extended abstract.

Observation : A supported empty regular corridor satisfies conditions C1 and C2 in Section 2.

4.1 Computing the optimal solution

We associate a direction of traversal with regular corridor. Each regular corridor (resp. link) will give rise to two *directed* regular corridors (resp. links). For brevity, from now on all regular corridor is assumed to be supported and directed. The angle of a link is defined to be the angle that its direction makes with the x-axis (between $-\pi$ and π). For all $p_i, p_j \in S$, define *link*(p_i, p_j, θ), $-\pi \leq \theta \leq \pi$, to be the 1-link corridor (possibly non-empty) that has an angle equal to θ and is supported by p_i and p_j. Cut *link*(p_i, p_j, θ) into two halves along $\overline{p_ip_j}$. Use *link*$^+$(p_i, p_j, θ) (resp. *link*$^-$(p_i, p_j, θ)) to denote the half that is after (resp. before) $\overline{p_ip_j}$ in the direction of *link*(p_i, p_j, θ). Define $C_\ell(p_i, p_j, \theta)$, $-\pi \leq \theta \leq \pi$, to be a *widest almost empty* regular corridor with ℓ links such that its last link has angle θ and is supported by p_i and p_j. $C_\ell(p_i, p_j, \theta)$ is almost empty if it does not contain any point of S after we remove the largest unbounded rectangle contained in *link*$^+$(p_i, p_j, θ) (see Figure 4). Clearly, any empty regular corridor is also almost empty.

Refer to Figure 4. Take some $C_{\ell-1}(p_k, p_l, \beta)$ and *link*(p_i, p_j, θ) such that *link*(p_i, p_j, θ) is perpendicular to and intersects *link*(p_k, p_l, β). Let *link*(p_i, p_j, θ) $\cap$ *link*(p_k, p_l, β) be R (a rectangular region). We can obtain a regular corridor with ℓ links as follows: remove the portion of *link*$^+$(p_k, p_l, β) after R, remove the portion of *link*$^-$(p_i, p_j, θ) before R, and then concatenate the remains of $C_{\ell-1}(p_k, p_l, \beta)$ and *link*(p_i, p_j, θ). If the corridor obtained still contains p_i, p_j, p_k, and p_l on its boundary,

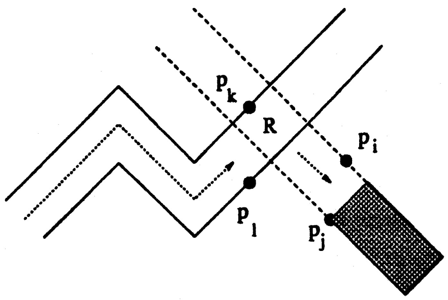


Figure 4. The solid lines represent $C_{\ell-1}(p_k, p_l, \beta)$. The dashed lines represent $link(p_i, p_j, \theta)$. The shaded region is the largest unbounded rectangle contained in $link^{+}(p_i, p_j, \theta)$.

then we define it to be $C_{\ell-1}(p_k, p_l, \beta) \oplus link(p_i, p_j, \theta)$. Otherwise, $C_{\ell-1}(p_k, p_l, \beta) \oplus link(p_i, p_j, \theta)$ is undefined.

Define $\lambda(p_i, p_j, \theta)$ to be the width of $link(p_i, p_j, \theta)$. $\lambda(p_i, p_j, \theta)$ is a sinusoidal function in θ with a constant phase difference dependent on p_i and p_j.[4] Define $\mathcal{A}$ to be the set of all $\lambda(*, *, \theta + \Delta)$'s where $\Delta = 0, \pm\pi/2, \pm 3\pi/2$. Let $I(\mathcal{A})$ be the set of all intersections of functions in $\mathcal{A}$ within the range $[-\pi, \pi]$. For each p_i and p_j, let α_{ij} be the nonnegative angle that the line through p_i and p_j makes with the x-axis. Define $\mathcal{B}$ to be $[-\pi, \pi] \cap \{\alpha_{ij} + \Delta : \Delta = 0, \pm\pi/2, \pm\pi, \pm 3\pi/2, \pm 2\pi\}$. Define $w_\ell(p_i, p_j, \theta)$ to be the width of $C_\ell(p_i, p_j, \theta)$. For any particular θ, we take $w_\ell(p_i, p_j, \theta)$ to be zero if $C_\ell(p_i, p_j, \theta)$ does not exist.

Observation : $|\mathcal{A}| = O(n^2)$, $|\mathcal{B}| = O(n^2)$, and $|I(\mathcal{A})| = O(n^4)$.

Lemma 3 *Given any $\ell > 1$ and $\theta \in [-\pi, \pi]$, $C_\ell(p_i, p_j, \theta)$ has the same width as $C_{\ell-1}(p_k, p_l, \beta) \oplus link(p_i, p_j, \theta)$ for some $C_{\ell-1}(p_k, p_l, \beta)$.*

Proof Set p_k and p_l to be the two points supporting the last but one link of $C_\ell(p_i, p_j, \theta)$ and set β to be the angle of that link. □

Lemma 4 *Given p_i and p_j, $w_\ell(p_i, p_j, \theta)$ is a possibly discontinuous piecewise sinusoidal function in θ on $[-\pi, \pi]$. The sinusoidal pieces are arc segments of functions in $\mathcal{A}$ and discontinuities can only occur at values in $\mathcal{B}$.*

Proof (Sketch) By induction on ℓ. Consider $\ell = 1$. $w_1(p_i, p_j, \theta) = \lambda(p_i, p_j, \theta)$ or zero depending on whether $link(p_i, p_j, \theta)$ is almost empty or not. And the change can only occur at those values of θ such that the bounding line through p_i or p_j passes through another point of S. Those values of θ clearly belongs to $\mathcal{B}$ and this proves the basis step.

Assume that the theorem is true for $w_{\ell-1}(p_k, p_l, \theta)$ for all p_k and p_l and $\ell > 1$. By Lemma 3, we can first compute the width function, denoted by $\mu_{p_k p_l}(p_i, p_j, \theta)$, of $C_{\ell-1}(p_k, p_l, \beta) \oplus link(p_i, p_j, \theta)$ for all p_k and p_l. We take $\mu_{p_k p_l}(p_i, p_j, \theta)$ to be zero if $C_{\ell-1}(p_k, p_l, \beta) \oplus link(p_i, p_j, \theta)$ is undefined or not almost empty. Then $w_\ell(p_i, p_j, \theta)$ is the upper envelope of the $\mu_{**}(p_i, p_j, \theta)$'s.

If $C_{\ell-1}(p_k, p_l, \beta) \oplus link(p_i, p_j, \theta)$ is defined, then $\mu_{p_k p_l}(p_i, p_j, \theta) = \min\{w_{\ell-1}(p_k, p_l, \beta), \lambda(p_i, p_j, \theta)\}$. Consider the range $[0, \pi/2]$ for θ. β must be equal to $\theta + \Delta$ for some $\Delta = \pm\pi/2$. To find $\mu_{p_k p_l}(p_i, p_j, \theta)$ within $[0, \pi/2]$, we should shift $w_{\ell-1}(p_k, p_l, \beta)$ by $-\Delta$ and then find the lower envelope of it and $\lambda(p_i, p_j, \theta)$ within $[0, \pi/2]$. By induction assumption, introducing a phase difference of $\pm\pi/2$ to the sinusoidal pieces of $w_{\ell-1}(p_k, p_l, \beta)$ will generate sinusoidal pieces of some other functions in $\mathcal{A}$. Therefore, the sinusoidal pieces of $\mu_{p_k p_l}(p_i, p_j, \theta)$ within $[0, \pi/2]$ are also arc segments of functions in $\mathcal{A}$. The other three ranges for θ can be handled similarly. We analyze the discontinuities in $\mu_{p_k p_l}(p_i, p_j, \theta)$ below.

For brevity, when a discontinuity occurs at a value in $\mathcal{B}$, we say that it is in $\mathcal{B}$. As θ is varied, if $C_{\ell-1}(p_k, p_l, \beta)$ is defined, then a change in the status of $C_{\ell-1}(p_k, p_l, \beta) \oplus link(p_i, p_j, \theta)$ being defined or almost empty may take place when a bounding line of $link(p_i, p_j, \theta)$ or $link(p_k, p_l, \beta)$ is about to include or lose a point of S. In the first case, θ is clearly in $\mathcal{B}$. In the second case, β is in $\mathcal{B}$ and so is θ as they differ by $\pm\pi/2$ or $\pm 3\pi/2$. The remaining possibility is that $C_{\ell-1}(p_k, p_l, \beta)$ is not defined. That is, a discontinuity in $w_{\ell-1}(p_k, p_l, \beta)$ is shifted and becomes a discontinuity in $w_\ell(p_i, p_j, \theta)$. By inspecting the possible amounts of shifts for different values of m, it can be verified that the shifted discontinuities are still in $\mathcal{B}$.

Finally, since all $\mu_{**}(p_i, p_j, \theta)$'s satisfy the conditions of the lemma and their mutual intersections are intersections of functions in $\mathcal{A}$, their upper envelope that is $w_\ell(p_i, p_j, \theta)$ satisfies the conditions of the lemma too. □

The algorithm is a direct implementation of the inductive proof of Lemma 4. Computing $\mu_{p_k p_l}(p_i, p_j, \theta)$ involves finding the lower envelope of appropriate portions of $w_{\ell-1}(p_k, p_l, \beta)$ and $\lambda(p_i, p_j, \theta)$ and pulling the curve down to zero whenever $C_{\ell-1}(p_k, p_l, \beta) \oplus link(p_i, p_j, \theta)$ is undefined or not almost empty. By rotating $link(p_k, p_l, \beta)$ and $link(p_i, p_j, \theta)$, we can compute in $O(n)$ time the $O(n)$ intervals on the θ-axis within which $C_{\ell-1}(p_k, p_l, \beta) \oplus link(p_i, p_j, \theta)$ is undefined or not almost empty. Given these intervals, we can do a plane sweep of the appropriate portions of $w_{\ell-1}(p_k, p_l, \beta)$ and $\lambda(p_i, p_j, \theta)$ and obtain $\mu_{p_k p_l}(p_i, p_j, \theta)$ in $O(n^4)$ time.

[4] It is actually not a sinusoidal function since we take the absolute value. But we shall abuse the notation for convenience.

Then we can sweep the $O(n^2)$ $\mu_{**}(p_i, p_j, \theta)$'s and take $w_\ell(p_i, p_j, \theta)$ to be the upper envelope. This takes $O(n^6)$ time and hence a total of $O(n^8)$ time for the $O(n^2)$ $w_\ell(*, *, \theta)$'s.

Theorem 2 *The widest empty regular corridor of at most ℓ links can be computed in $O(\ell n^8)$ time.* □

4.2 An approximation algorithm

We first present a faster algorithm when the corridor is restricted to be orthogonal. The direction of each link must then be 0 or $\pi/2$ or π or $-\pi/2$. This will enable us to speed up the computation of each $w_\ell(p_i, p_j, \theta)$ in the algorithm described in Section 4.1. Suppose that $\theta = 0$ and consider $C_{\ell-1}(p_k, p_l, -\pi/2) \oplus link(p_i, p_j, 0)$. Let y_{kl} be the y-coordinate of the highest point of S lying in the open region bounded by $link^+(p_k, p_l, -\pi/2)$. Let x_{ij} be the x-coordinate of the rightmost point of S lying in the open region bounded by $link^-(p_i, p_j, 0)$. The following are necessary and sufficient conditions that the result of $\oplus$ will be defined and almost empty:

1. $x_{ij} \le \min\{x(p_k), x(p_l)\} \le \min\{x(p_i), x(p_j)\}$
2. $\max\{x(p_k), x(p_l)\} \le \max\{x(p_i), x(p_j)\}$
3. $\min\{y(p_k), y(p_l)\} \ge \min\{y(p_i), y(p_j)\} \ge y_{kl}$
4. $\max\{y(p_k), y(p_l)\} \ge \max\{y(p_i), y(p_j)\}$

By using an orthogonal range searching structure on S, y_{kl} and x_{ij} can be precomputed in $O(\log^2 n)$ time. We map each $C_{\ell-1}(p_k, p_l, -\pi/2)$ into the point $(\min\{x(p_k), x(p_l)\}, \max\{x(p_k), x(p_l)\}, \min\{y(p_k), y(p_l)\}, \max\{y(p_k), y(p_l)\}, y_{kl}, w_{\ell-1}(p_k, p_l, -\pi/2))$ in R^6. Using an appropriate multi-level tree structure to store these points, we can search in $O(\log^5 n)$ time to find the widest $C_{\ell-1}(p_k, p_l, -\pi/2)$ that satisfies conditions (1)–(4). The other orientations can be handled similarly.

Lemma 5 *The widest empty regular corridor with at most ℓ links can be computed in $O(\ell n^2 \log^5 n)$ time if it is restricted to be orthogonal.* □

Our approximation algorithm works as follows. We choose a real number $\alpha > 0$ and draw $\lfloor 4\pi/\alpha \rfloor$ directed lines through the origin. Then we treat each directed line as the y-axis and apply Lemma 5. Finally, we select the widest corridor among those computed. The following theorem shows that α can be chosen such that the width of the corridor computed can be made arbitrarily close to the optimal.

Theorem 3 *For any $\epsilon > 0$, an empty regular corridor with at most ℓ links and width at least $(1-\epsilon)$ times the optimal can be computed in $O((1/\epsilon)\ell n^3 \log^5 n)$ time.*

Proof (Sketch) We first shrink/expand S so that the smallest enclosing square of S has side length equal to 1. Suppose that we have drawn the directed lines as described above. The value of α will be fixed later. Let C be a widest empty regular corridor with at most ℓ links. By Lemma 2, we can assume that C is supported. We apply a clockwise rotational deformation to C until either some link of C is parallel to some directed line or the width of some link of C becomes zero. Emptiness is maintained throughout the rotation. When a link is rotated, we always take the pivots to be the current rightmost point of S on the lower boundary and the current leftmost point of S on the upper boundary. Let A be a link with the minimum width when the rotational deformation stops. Consider some moment of the rotational deformation at which A began to rotate about different pivots. Let α' be the angle that A would turn when A would rotate about different pivots again ($\alpha' \le \alpha$). It can be shown that the decrease in width in this stage is at most $\sqrt{2}\alpha'$. Generalizing this argument, we get an upper bound of $\sqrt{2}\alpha$ on the total decrease in width and hence the error of approximation. By the pigeonhole principle, there exists a horizontal or vertical empty 1-link corridor with width at least $1/(n-1)$. Therefore, the error ratio is at most $\sqrt{2}(n-1)\alpha$. The theorem follows if we set α to be $\epsilon/\sqrt{2}(n-1)$. □

References

[1] S. Chattopadhyay and P. Das, *The k-dense corridor problems*, Pattern Recognition Letters, 11 (1990), pp. 463–469.

[2] M. Houle and A. Maciel, *Finding the widest empty corridor through a set of points*, in Snapshots of computational and discrete geometry, G. Toussaint, ed., pp. 201–213.

[3] R. Janardan and F. Preparata, *Widest-corridor problems*, Tech. Report TR 93-17, Department of Computer Science, University of Minnesota (Twin Cities), March 1993. See also Proceedings of the Fifth Canadian Conference on Computational Geometry, pp. 426–431, 1993.

[4] N. Sarnak and R. Tarjan, *Planar point location using persistent search trees*, Communications of the ACM, 29 (1986), pp. 669–679.