

Constructing Pairwise Disjoint Paths with Few Links

Himanshu Gupta¹ and Rephael Wenger^{2*}

¹ Stanford University, Stanford CA 94305 (hgupta@cs.stanford.edu)

² The Ohio State University, Columbus, OH 43210 (wenger@cis.ohio-state.edu).

Abstract. Let P be a simple polygon and let $\{(u_i, u'_i)\}$ be m pairs of distinct vertices of P where for every distinct $i, j \leq m$, there exist pairwise disjoint paths connecting u_i to u'_i and u_j to u'_j . We wish to construct m pairwise disjoint paths in the interior of P connecting u_i to u'_i for $i = 1, \dots, m$, with minimal total number of line segments. We give an approximation algorithm which in $O(n \log m + M \log m)$ time constructs such a set of paths using $O(M)$ line segments where M is the number of line segments in the optimal solution.

1 Introduction

Let P be a simple polygon and let u and u' be two distinct vertices of P . The (*interior*) *link distance* from u to u' is the minimum number of line segments (also called *links*) required to connect u to u' by a polygonal path lying in (the interior of) P . The interior link distance from u to u' may differ greatly from the link distance between the two points. (See Figure 1.) A polygonal path which uses the minimum number of required line segments is called a *minimum link (interior) path*. Suri in [11] gave a linear time algorithm for determining the link distance and a minimal link path between two vertices.

Let u_1, u'_1, u_2, u'_2 be four vertices lying in the given order around P . By virtue of the relative locations of these four vertices, there are nonintersecting paths, ζ_1 and ζ_2 , connecting u_1 to u'_1 and u_2 to u'_2 , respectively. However, it is possible that every minimum interior link path connecting u_1 to u'_1 intersects every minimum interior link path connecting u_2 to u'_2 . (See Figure 1.) To simultaneously connect u_1 to u'_1 and u_2 to u'_2 by nonintersecting interior paths requires more line segments. In general, two additional line segments suffice to construct two such nonintersecting interior paths. (See [7].)

A set $\Pi = \{(u_i, u'_i)\}$, $i \leq m$, of m pairs of distinct vertices of P is *untangled* if some set of pairwise disjoint paths connects each u_i to u'_i . Let $\Pi = \{(u_i, u'_i)\}$, $i \leq m$, be an untangled set of m pairs of distinct vertices of P . Let $l(u_i, u'_i)$ be the interior link distance from u_i to u'_i and let $L = \sum_{i=1..m} l(u_i, u'_i)$ be the sum of those distances. Clearly, L line segments are required to construct a set of pairwise disjoint interior paths connecting u_i to u'_i , for $i = 1, \dots, m$. How many additional line segments are required? In [7] we proved that $O(m \log m)$

* Supported by NSA grant MDA904-97-1-10019.

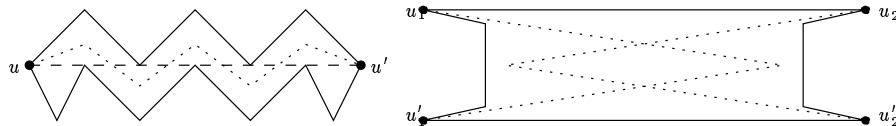


Fig. 1. Minimum link paths, minimum link interior paths and intersecting link paths.

additional line segments suffice and claimed without proof that $\Omega(m \log m)$ additional line segments may be required. A proof of the lower bound is provided in [6].

We define the *pairwise disjoint link paths problem* as: given an untangled set, $\{(u_i, u'_i)\}$, of m pairs of distinct vertices of P , find the minimum total number of line segments required by a set of pairwise disjoint interior paths connecting u_i to u'_i . We were unable to give a polynomial time algorithm for this problem or to determine if the problem is NP-complete. Instead we present an algorithm which finds a solution within a constant factor of the optimal solution. Related problems are shown to be NP-complete in [2] and [5], but we do not know if those results can be applied to our problem.

A triangulation T_P of P (possibly with interior vertices) is *isomorphic* to a triangulation T_Q of Q if there is a one-to-one, onto mapping f between the vertices of T_P and the vertices of T_Q such that p, p', p'' are vertices of a triangle in T_P if and only if $f(p), f(p'), f(p'')$ are vertices of a triangle in T_Q . An isomorphic triangulation of P and Q defines a piecewise linear homeomorphism between P and Q . The size of a triangulation is the total number of vertices, edges and triangles in the triangulation.

Algorithms for constructing isomorphic triangulations and piecewise linear homeomorphisms between simple polygons are also given in [1,8,7]. Algorithms for constructing isomorphic triangulations between labelled point sets are described in [9] and [10]. The main result in this paper improves the output size and running time of the approximation algorithm in [7] from $O(M_1 \log n + n \log^2 n)$ to $O(M_1 \log n)$ where n is the input size and M_1 is the size of the optimal solution. The improvement is described in [6].

2 Approximation Algorithm

We first give an approximation algorithm for connecting a set of vertices \mathcal{U} by pairwise disjoint interior paths to a distinguished edge e^* of P . We start with some definitions.

Point $p \in P$ is *visible* from point $p' \in P$ if P contains the open line segment (p, p') . Point p is *clearly visible* from point $p' \in P$ if the interior of P contains the open line segment (p, p') .

Point $p \in P$ is *(clearly) visible* from edge $e \in P$ if there is some point $p' \in e$ such that p is (clearly) visible from p' . (This definition of visibility is sometimes called *weak visibility* as opposed to *strong visibility* where p must be visible from

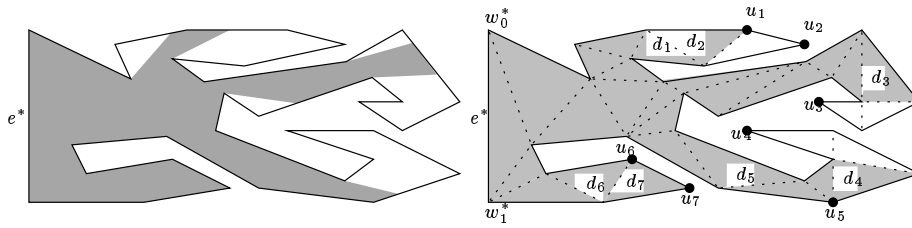


Fig. 2. $\hat{\text{Vis}}(e)$ and Γ_{e^*} .

every point $p' \in e$. Throughout this paper, visibility refers to weak visibility.) Edge e or triangle t is (clearly) visible from edge e' or triangle t' if there are points $p \in e$ or $p \in t$ and $p' \in e'$ or $p' \in t'$ such that p is (clearly) visible from p' .

We let $\hat{\text{Vis}}(p)$ and $\hat{\text{Vis}}(e)$ denote the points clearly visible from point p and edge e . (See Figure 2.) Note that $\hat{\text{Vis}}(p)$ and $\hat{\text{Vis}}(e)$ are not necessarily closed sets.

Let u and e be a vertex and an edge of P , respectively. Edge d of triangulation T_P separates u from e if every interior path from u to the interior of e must intersect the interior of d . Triangle t of triangulation T_P separates u from e if every interior path from u to the interior of e must intersect the interior of t .

To construct pairwise disjoint paths connecting the vertices \mathcal{U} to edge e^* of P , we construct a triangulated region Γ_{e^*} which contains and approximates $\hat{\text{Vis}}(e^*)$, the set of points clearly visible from e^* . For each $u_i \in \mathcal{U}$, let d_i be the diagonal of Γ_{e^*} farthest from e^* which separates e^* from u_i . Let s_i be the portion of e^* visible from d_i . Note that s_i is a line segment. (See Figures 2 and 3.)

For each u_i we wish to choose a point p_i on e^* to be the endpoint of the path from u_i to e^* . Obviously, a point in s_i is a good candidate since it can reach d_i with a single line segment. However, we also need to choose the p_i such that their order on e^* is consistent with the order of \mathcal{U} around P . In other words, u_i, u_j, p_j, p_i should lie clockwise or counter-clockwise around P in the given order.

We partition the set of line segments $\{s_i\}$ into groups and associate each such group with a point g_j on e^* which is in the “middle” of the line segments in the groups. If many of the line segments in the group contain g_j , then the corresponding diagonals can be connected to g_j by pairwise disjoint line segments. If few line segments contain g_j , then g_j partitions the line segments into roughly two equal subgroups in e^* with the property that many line segments connecting d_i to s_i from one subgroup intersect many line segments connecting d_j to s_j from the other subgroup. In addition, the order that the points g_j lie on e^* is consistent with the order that the associated vertices of \mathcal{U} lie on the boundary of P . Partitioning the line segments is conceptually and technically the most difficult part of the algorithm.

From all the $\hat{\text{Vis}}(g_j)$, we construct another triangulated region $\Gamma \subseteq \Gamma_{e^*}$. We recursively connect \mathcal{U} by pairwise disjoint paths to the edges on the boundary of

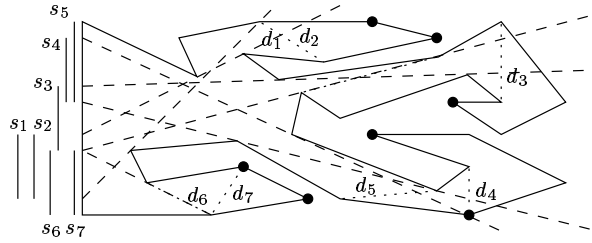


Fig. 3. Line segments $s_i = \text{int}(e^*) \cap \hat{V}\text{is}(d_i)$.

Γ and then connect those boundary edges by pairwise disjoint line segments to Γ . A careful analysis shows that Γ must contain many line segments in any set of pairwise disjoint paths connecting \mathcal{U} to e^* . Thus the number of line segments in our solution is proportional to the number in the optimal solution.

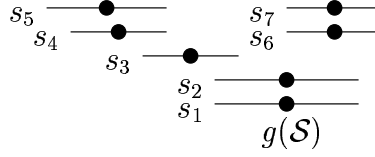
Lemma 1. *Let P be a simple polygon on n vertices with distinguished edge $e^* = \{w_0^*, w_1^*\}$ and let \mathcal{U} be a subset of $\text{Vert}(P) \setminus \{w_0^*, w_1^*\}$ of size m . A set of m pairwise disjoint interior paths connecting the vertices in \mathcal{U} to the interior of e^* can be constructed in $O(n \log m + M \log m)$ time using a total of at most $240M$ line segments where M is the minimum total number of line segments necessary to connect \mathcal{U} to e^* by m pairwise disjoint paths.*

Proof. Let u_1, u_2, \dots, u_m be the points in \mathcal{U} labeled in clockwise order around P starting at e^* . Construct a triangulation T_P of P . Let Γ_{e^*} be union of the triangles of T_P which are clearly visible from edge e^* . The region Γ_{e^*} is a simple polygon in P . (See Figure 2.)

For each $u_i \in \mathcal{U}$, let d_i be the diagonal of Γ_{e^*} farthest from e^* which separates e^* from u_i . Let s_i be $\text{int}(e^*) \cap \hat{V}\text{is}(d_i)$, the interior of e^* which is clearly visible from s_i . Set s_i is an open line segment lying on e^* . Note that s_i may equal s_j (and d_i may equal d_j) for many distinct points $u_i, u_j \in \mathcal{U}$. (See Figure 3.)

Let \mathcal{S} be any set of open line segments in \mathbf{R}^1 , not necessarily distinct. For each point $q \in \mathbf{R}^1$, let $f(q, \mathcal{S})$ be the number of line segments of \mathcal{S} which contain the point q . The line segments of \mathcal{S} are open and do not contain their endpoints. Let $f^-(q, \mathcal{S})$ and $f^+(q, \mathcal{S})$ be the number of line segments of \mathcal{S} contained in the open intervals $(-\infty, q)$ and (q, ∞) , respectively. Note that $f(q, \mathcal{S}) + f^-(q, \mathcal{S}) + f^+(q, \mathcal{S})$ equals $|\mathcal{S}|$.

Let \mathcal{R} be the set of midpoints of line segments of \mathcal{S} , again not necessarily distinct. The median point of \mathcal{R} is the $\lceil |\mathcal{R}|/2 \rceil$ 'th point in \mathcal{R} , ordered from $-\infty$ to ∞ . Let $g(\mathcal{S})$ be this median point of \mathcal{R} . At least $\lceil |\mathcal{R}|/2 \rceil = \lceil |\mathcal{S}|/2 \rceil$ points of \mathcal{R} lie in each of the closed intervals $(-\infty, g(\mathcal{S})]$ and $[g(\mathcal{S}), \infty)$. If the midpoint of segment $s \in \mathcal{S}$ lies in $(-\infty, g(\mathcal{S})]$, then segment s either contains $g(\mathcal{S})$ or lies in the open interval $(-\infty, g(\mathcal{S}))$. Thus $f(g(\mathcal{S}), \mathcal{S}) + f^-(g(\mathcal{S}), \mathcal{S})$ is greater than or equal to $\lceil |\mathcal{S}|/2 \rceil$. Similarly $f(g(\mathcal{S}), \mathcal{S}) + f^+(g(\mathcal{S}), \mathcal{S})$ is greater than or equal to $\lceil |\mathcal{S}|/2 \rceil$. (See Figure 4.)



$$f(g(\mathcal{S}), \mathcal{S}) = 2, f^-(g(\mathcal{S}), \mathcal{S}) = 3, f^+(g(\mathcal{S}), \mathcal{S}) = 2.$$

Fig. 4. \mathcal{S} , $g(\mathcal{S})$ and $f(g(\mathcal{S}), \mathcal{S})$.

Now consider two sets of line segments \mathcal{S}_0 and \mathcal{S}_1 on \mathbf{R}^1 and let $\mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_1$. Define

$$\mathcal{F}(\mathcal{S}_0, \mathcal{S}_1) = f(g(\mathcal{S}), \mathcal{S}) + f^+(g(\mathcal{S}), \mathcal{S}_0) + f^-(g(\mathcal{S}), \mathcal{S}_1).$$

Without loss of generality, assume that w_0^*, e^*, w_1^* appear in counter-clockwise order around P . Let $\mathcal{S}_{\mathcal{U}}$ be the sequence (s_1, s_2, \dots, s_m) . Embed e^* and the line segments $s_i \in \mathcal{S}$ in the real line \mathbf{R}^1 , mapping w_0^* to zero and w_1^* to one. In the next section, we describe an algorithm to partition $\mathcal{S}_{\mathcal{U}}$ into contiguous subsequences $\sigma_1 = (s_1, s_2, \dots, s_{i_1})$, $\sigma_2 = (s_{i_1+1}, s_{i_1+2}, \dots, s_{i_2})$, \dots , $\sigma_{2h} = (s_{i_{2h-1}+1}, s_{i_{2h-1}+2}, \dots, s_m)$, such that:

1. $g(\sigma_1 \cup \sigma_2) \leq g(\sigma_3 \cup \sigma_4) \leq \dots \leq g(\sigma_{2h-1} \cup \sigma_{2h})$;
2. $|\sigma_{2j-1}| = |\sigma_{2j}| + 1$ for $1 \leq j \leq h$;
3. $\sum_{j=1..h} \mathcal{F}(\sigma_{2j-1}, \sigma_{2j}) \geq m/40$.

(One possible partition of the segments in Figure 3 is $\sigma_1 = \{s_1, s_2, s_3\}$, $\sigma_2 = \{s_4, s_5\}$, $\sigma_3 = \{s_6\}$, $\sigma_4 = \{s_7\}$.)

Let g_j equal $g(\sigma_{2j-1} \cup \sigma_{2j})$ for $j = 1, \dots, h$. Note that g_1, g_2, \dots, g_h lie in counter-clockwise order around P . Let $\mathcal{U}_j = \{u_i : s_i \in \sigma_j\}$ be the points in \mathcal{U} corresponding to the line segments in σ_j for $j = 1, \dots, h$. For each g_j , let Γ_j be the union of the triangles of T_P which intersect $\hat{\text{Vis}}(g_j)$ and separate some $u \in \mathcal{U}_{2j-1} \cup \mathcal{U}_{2j}$ from e^* . (See Figure 5.) Let Γ be the union of all the Γ_j . Similar to Γ_{e^*} , the region Γ is also a simple polygon in P , its boundary is composed of edges and chords of P , and it has a triangulation T_Γ induced by the triangulation T_P of P .

Let \mathcal{C} be the set of chords of P bounding Γ . Each chord $c \in \mathcal{C}$ separates P into two subpolygons. Let P_c be the subpolygon not containing Γ . Let w_0^c and w_1^c be the endpoints of c . For each chord $c \in \mathcal{C}$, let \mathcal{U}_c be the points of $\mathcal{U} \setminus \{w_0^c, w_1^c\}$ in P_c . Recursively, construct pairwise disjoint paths connecting the points in \mathcal{U}_c to c . (See Figure 6.)

For each $u_i \in \mathcal{U}$, let \tilde{d}_i be the diagonal of Γ farthest from e^* which separates e^* from u_i . Choose the minimum j such that \tilde{d}_i is a diagonal of Γ_j . Connect \tilde{d}_i to g_j by a line segment λ_i in the interior of P . (See Figure 7.) Diagonal \tilde{d}_i may also separate other vertices of \mathcal{U} from e^* and there may be many line segments which intersect \tilde{d}_i . The line segments λ_i should be chosen so that their order along \tilde{d}_i corresponds to the order of the vertices around P . The choice of \tilde{d}_i

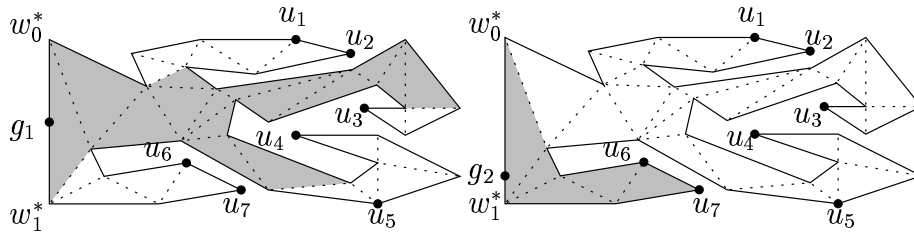


Fig. 5. Γ_1 and Γ_2 .

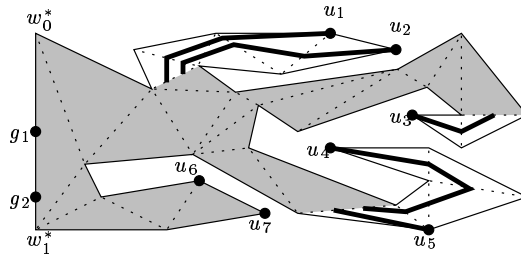


Fig. 6. Γ , triangulation T_Γ and paths to the boundary of Γ .

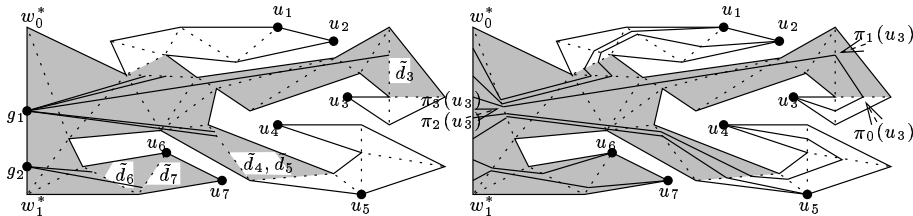


Fig. 7. Diagonals \tilde{d}_i , line segments λ_i and paths connecting \mathcal{U} to e^* .

and the associated point g_j ensures that line segments λ_i intersect only at their endpoints. (See [6].)

For each point $u_i \in \mathcal{U}(c)$, let $\pi_0(u_i)$ be the endpoint on c of the path connecting u_i to c . For each point $u_i \in \mathcal{U}$ which lies in Γ , let $\pi_0(u_i)$ equal u_i . Let $\pi_1(u_i)$ be the endpoint of λ_i on \tilde{d}_i . Let $\pi_2(u_i)$ be the first intersection point of λ_i and the triangle containing e^* . Place m points equally spaced on e^* . Let $\pi_3(u_i)$ be the i 'th point, ordered counter-clockwise from w_0^* . Connect $\pi_0(u_i)$ to e^* with a polygonal line through $\pi_0(u_i), \pi_1(u_i), \pi_2(u_i), \pi_3(u_i)$. (See Figure 7.)

We claim that this algorithm connects \mathcal{U} to e^* using $O(M)$ links where M is the number of links in some optimal solution. For each $u_i \in \mathcal{U}$, let ζ_i be the path constructed from u_i to e^* by our algorithm while η_i is the path from u_i to e^* in the optimal solution. Path ζ_i has at most three line segments in Γ . Line segment s_i is in $\sigma_{2j-1} \cup \sigma_{2j}$ for some j . If s_i contains g_j , then some point on

diagonal d_i is clearly visible from g_j and d_i is a diagonal of $\Gamma_j \subseteq \Gamma$. Since d_i is the farthest diagonal visible from e^* which separates u_i from e^* , any path from u_i to e^* must have at least one line segment contained in $\Gamma_j \subseteq \Gamma$. Thus if s_i contains g_j , then we can charge the three links of ζ_i in Γ to a line segment of η_i in Γ . However, s_i may not contain g_j .

Consider the case where $s_i \in \sigma_{2j-1}$ lies between g_j and w_1^* while $s_{i'} \in \sigma_{2j}$ lies between w_0^* and g_j . Any two paths from u_i to s_i and $u_{i'}$ to $s_{i'}$ must intersect. Since paths η_i and $\eta_{i'}$ are pairwise disjoint, either the endpoint of η_i must lie between w_0^* and g_j or the endpoint of $\eta_{i'}$ must lie between g_j and w_1^* . Without loss of generality, assume that the endpoint ρ of η_i lies between w_0^* and g_j . In that case, g_j lies between ρ and s_i . Let d be the farthest diagonal of P visible from ρ and separating ρ from u_i . By the construction of d_i and s_i , diagonal d separates d_i from e^* and hence is visible to s_i . Since g_j lies between ρ and s_i , diagonal d is also visible to g_j and is contained in $\Gamma_j \subseteq \Gamma$. Thus if ρ lies between w_0^* and g_j , path η_i must have at least one line segment contained in $\Gamma_j \subseteq \Gamma$. Similarly, if the endpoint of $\eta_{i'}$ lies between g_j and w_1^* , path $\eta_{i'}$ must have at least one line segment contained in $\Gamma_j \subseteq \Gamma$. It follows that either η_i or $\eta_{i'}$ must have a line segment contained in $\Gamma_j \subseteq \Gamma$.

Let m_0, m_-, m_+ equal $f(g_j, \sigma_{2j-1}), f^-(g_j, \sigma_{2j-1})$ and $f^+(g_j, \sigma_{2j-1})$, respectively, while m'_0, m'_-, m'_+ equal $f(g_j, \sigma_{2j}), f^-(g_j, \sigma_{2j})$ and $f^+(g_j, \sigma_{2j})$, respectively. By the arguments above, the paths connecting the points in $\mathcal{U}_{2j-1} \cup \mathcal{U}_{2j}$ to e^* in the optimal solution must have at least $m_0 + m'_0 + \min(m_+, m'_-)$ line segments contained in Γ . By the choice of point g_j , $m_0 + m_- + m'_0 + m'_- \geq |\sigma_{2j-1} \cup \sigma_{2j}|/2$. Since $|\sigma_{2j-1}|$ equals $|\sigma_{2j}|$ or $|\sigma_{2j-1}| + 1$, $m_0 + m_- + m'_0 + m'_- \geq |\sigma_{2j-1}|$. On the other hand, $m_0 + m_- + m_+ = |\sigma_{2j-1}|$. Subtracting the second equation from the first gives $m'_0 + m'_- \geq m_+$. Thus

$$\begin{aligned} m_0 + m'_0 + \min(m_+, m'_-) &= \min(m_0 + m'_0 + m_+, m_0 + m'_0 + m'_-) \\ &\geq \min(m_0 + m'_0 + m_+, m_0 + m_+) \\ &= m_0 + m_+ \end{aligned}$$

Similarly, $m_0 + m_+ \geq m'_-$ and $m_0 + m'_0 + \min(m_+, m'_-) \geq m'_0 + m'_-$. Thus

$$m_0 + m'_0 + \min(m_+, m'_-) \geq \max(m_0 + m_+, m'_0 + m'_-) \geq \mathcal{F}(g_j, \sigma_{2j-1}, \sigma_{2j})/2.$$

The paths connecting the points in $\mathcal{U}_{2j-1} \cup \mathcal{U}_{2j}$ to e^* in the optimal solution must have at least $\mathcal{F}(g_j, \sigma_{2j-1}, \sigma_{2j})/2$ line segments in Γ . Since $\sum_{j=1..h} \mathcal{F}(\sigma_{2j-1}, \sigma_{2j}) \geq m/40$, any pairwise disjoint paths connecting the points in \mathcal{U} to e^* must have at least $m/80$ line segments contained in Γ . The construction produces at most $3m$ line segments in Γ , so the solution is at most 240 times the optimal.

Finally, we discuss the running time of our algorithm. Constructing the initial triangulation T_P takes $O(n)$ time [3]. As discussed in the next section, partitioning \mathcal{S}_U into the subsequences σ_j takes $O(m \log m)$ time. Constructing Γ_{e^*} takes $O(n^*)$ time where n^* is the number of triangles of T_P intersected by $\hat{\text{Vis}}(e^*)$ [4]. All the other steps in the algorithm can be done in $O(n^* + m)$ time. Thus the

```

PARTITION( $\mathcal{S}$ )
/*  $\mathcal{S}$  = a sequence of line segments  $(s_1, s_2, \dots, s_m)$  */
/* Returns a linked list of contiguous subsequences of  $\mathcal{S}$  */
1. Initialize linked list  $\mathcal{A}$  to  $\emptyset$ ;
2. FOR  $i = 1$  TO  $m$  DO
3.     Create new node  $a$  where  $a.seq = (s_i)$  and  $a.size = 1$ ;
4.     Add  $a$  to the end of linked list  $\mathcal{A}$ ;
5. WHILE  $\exists a \in \mathcal{A}$  such that  $g(a.seq) > g(a.next.seq)$  DO
6.     Merge  $a$  and  $a.next$  to form a new node  $a'$  in  $\mathcal{A}$ ;
7.     BALANCE-NEXT( $a'$ );
8.     BALANCE-PREV( $a'$ );
9. Return( $\mathcal{A}$ ).

```

Fig. 8. Algorithm PARTITION.

non-recursive steps in this algorithm take $O(n^* + m \log m)$ time. A careful accounting for the recursive steps gives the desired $O(n \log m + M \log m)$ bound. Details appear in [6]. \square

Using arguments similar to those given in [7], the previous algorithm can be turned into an algorithm for connecting an untangled set of m pairs of vertices of P . The algorithm and its analysis is provided in [6].

Theorem 2. *Let P be a simple polygon on n vertices let $\Pi = \{(u, u')\}$ be an untangled set of m pairs of distinct vertices of P . A set of m pairwise disjoint interior paths connecting u to u' for each $(u, u') \in \Pi$ can be constructed in $O(n \log m + M \log m)$ time using $O(M)$ line segments where M is the minimum total number of line segments necessary to connect all pairs $(s, s') \in \Pi$ by pairwise disjoint paths.*

3 Partition Algorithm

In this section, we describe and analyze the algorithm for partitioning a sequence of line segments. The functions f , f^+ , f^- and \mathcal{F} were defined in the previous section.

Lemma 3. *Let \mathcal{S} be a sequence of line segments (s_1, s_2, \dots, s_m) on the real line \mathbf{R}^1 . In $O(m \log m)$ time, \mathcal{S} can be partitioned into contiguous subsequences $\sigma_1 = (s_1, s_2, \dots, s_{i_1})$, $\sigma_2 = (s_{i_1+1}, s_{i_1+2}, \dots, s_{i_2})$, \dots , $\sigma_h = (s_{i_{h-1}+1}, s_{i_{h-1}+2}, \dots, s_m)$, such that:*

1. $g(\sigma_1 \cup \sigma_2) \leq g(\sigma_3 \cup \sigma_4) \leq \dots \leq g(\sigma_{2h-1} \cup \sigma_{2h})$;
2. $|\sigma_{2j-1}| = |\sigma_{2j}|$ (+1) for $1 \leq j \leq h$;
3. $\sum_{j=1..h} \mathcal{F}(\sigma_{2j-1}, \sigma_{2j}) \geq m/40$.

Proof (outline). Split \mathcal{S} into m distinct subsequences, (s_i) , consisting of one element each. Store the m subsequences in a linked list \mathcal{A} in the order they appear in \mathcal{S} . Each node $a \in \mathcal{A}$ contains a subsequence $a.seq$. Call \mathcal{A} *balanced* if the size of each subsequence is at most three times the size of any adjacent subsequence in \mathcal{A} .

While \mathcal{A} contains two adjacent subsequences, $a.seq$ followed by $a.next.seq$, such that $g(a.seq) > g(a.next.seq)$, merge the subsequences $a.seq$ and $a'.seq$. After each merge of two such subsequences, rebalance list \mathcal{A} by merging adjacent subsequences, as necessary. Figure 8 contains the main algorithm. A complete description of the subroutines BALANCE-NEXT and BALANCE-PREV is provided in [6].

Let a_j be the j 'th node in \mathcal{A} when the algorithm is completed. Partition $a_j.seq = (s_i, \dots, s_{i'})$ into two approximately equal sized sequences $\sigma_{2j-1} = (s_i, \dots, s_{\lceil(i+i')/2\rceil})$ and $\sigma_{2j} = (s_{\lceil(i+i')/2\rceil+1}, \dots, s_{i'})$. We claim that this is a partitioning of \mathcal{S} with the desired properties. Initially, the s_j are stored in \mathcal{A} in sorted order. The merging and splitting steps in the main algorithm and in the subroutines BALANCE-NEXT and BALANCE-PREV preserve the order of the s_i , so the σ_j properly partition \mathcal{S} into contiguous subsequences.

Let g_j be $g(a_j.seq) = g(\sigma_{2j-1} \cup \sigma_{2j})$. The while loop only terminates when $g_1 \leq g_2 \leq \dots \leq g_h$, so property 1 is clearly satisfied. Sets σ_{2j-1} and σ_{2j} are created by partitioning $a_j.seq$ into two equal sized sequences, so property 2 is satisfied.

To show property 3 holds, note that a_j could be an initial node or it could be created when $g(a.seq) > g(a.next.seq)$ or it could be created in the rebalancing procedure.

If a_j is an initial node, then $a_j.seq = \{s\}$ for some $s \in \mathcal{S}$ and $\sigma_{2j-1} = \{s\}$ and $\sigma_{2j} = \emptyset$. Point g_j is the midpoint of s and $\mathcal{F}(\sigma_{2j-1}, \sigma_{2j}) \geq 1 \geq (1/8)|a_j|$.

Assume a_j is created when $g(a.seq)$ is greater than $g(a.next.seq)$ and that $|a.next| \geq |a|$. The sequence $a.seq$ is a subsequence of σ_{2j-1} , so

$$\begin{aligned} f(g_j, \sigma_{2j-1}) &\geq f(g_j, a.seq) \text{ and} \\ f^+(g_j, \sigma_{2j-1}) &\geq f^+(g_j, a.seq). \end{aligned}$$

The point $g_j = g(a_j.seq)$ must lie between $a.g$ and $a.next.g$, so

$$\begin{aligned} f(g_j, a.seq) + f^+(g_j, a.seq) &\geq f(g(a.seq), a.seq) + f^+(g(a.seq), a.seq) \\ &\geq |a|/2. \end{aligned}$$

Since $|a.next| \leq 3|a|$, we have $|a_j| \leq 4|a|$. Thus,

$$\mathcal{F}(\sigma_{2j-1}, \sigma_{2j}) \geq f(g_j, \sigma_{2j-1}) + f^+(g_j, \sigma_{2j-1}) \geq (1/8)|a_j|.$$

In the case that $|a.next| < |a|$, similar reasoning gives

$$\mathcal{F}(\sigma_{2j-1}, \sigma_{2j}) \geq f(g_j, \sigma_{2j}) + f^-(g_j, \sigma_{2j}) \geq (1/8)|a_j|.$$

Finally, if a_j is created in the rebalancing step, $\mathcal{F}(\sigma_{2j-1}, \sigma_{2j})$ may not have the desired lower bound. However, at most $(4/5)m$ line segments lie in nodes

created in the rebalancing step. By counting the $m/5$ line segments which are not in a rebalanced node, we find $\sum_{i=1..k} \mathcal{F}(\sigma_{2j-1}, \sigma_{2j}) \geq m/40$.

A complete description and analysis of the algorithm, its correctness and $O(m \log m)$ running time appears in [6]. \square

References

1. ARONOV, B., SEIDEL, R., AND SOUVAINE, D. On compatible triangulations of simple polygons. *Comput. Geom. Theory Appl.* 3, 1 (1993), 27–35.
2. BASTERT, O., AND FECKETE, S. Geometrische verdrahtungsprobleme. Technical Report 96.247, Angewandte Mathematik und Informatik, Universität zu Köln, Köln, Germany, 1996.
3. CHAZELLE, B. Triangulating a simple polygon in linear time. *Discrete Comput. Geom.* 6 (1991), 485–524.
4. GUIBAS, L. J., HERSHBERGER, J., LEVEN, D., SHARIR, M., AND TARJAN, R. E. Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. *Algorithmica* 2 (1987), 209–233.
5. GUIBAS, L. J., HERSHBERGER, J. E., MITCHELL, J. S. B., AND SNOEYINK, J. S. Approximating polygons and subdivisions with minimum link paths. *Internat. J. Comput. Geom. Appl.* 3, 4 (Dec. 1993), 383–415.
6. GUPTA, H., AND WENGER, R. Constructing pairwise disjoint paths with few links. Technical Report OSU-CISRC-2/97-TR16, The Ohio State University, Columbus, Ohio, 1997.
7. GUPTA, H., AND WENGER, R. Constructing piecewise linear homeomorphisms of simple polygons. *J. Algorithms* 22 (1997), 142–157.
8. KRANAKIS, E., AND URRUTIA, J. Isomorphic triangulations with small number of Steiner points. In *Proc. 7th Canad. Conf. Comput. Geom.* (1995), pp. 291–296.
9. SAALFELD, A. Joint triangulations and triangulation maps. In *Proc. 3rd Annu. ACM Sympos. Comput. Geom.* (1987), pp. 195–204.
10. SOUVAINE, D., AND WENGER, R. Constructing piecewise linear homeomorphisms. Technical Report 94–52, DIMACS, New Brunswick, New Jersey, 1994.
11. SURI, S. A linear time algorithm for minimum link paths inside a simple polygon. *Comput. Vision Graph. Image Process.* 35 (1986), 99–110.