Minimal Link Visibility Paths inside a Simple Polygon

Muhammad H. Alsuwaiyel[1] and D.T. Lee[2]
Department of Electrical Engineering and Computer Science
Northwestern University
Evanston, IL 60208

ABSTRACT We show that it is NP-hard to find a polygonal path π with a minimum number of turns inside a simple polygon P such that every point of P is visible from at least one point on π. In proving this main result, we show two other related problems to be NP-hard as well. Specifically, given a set S of points (edges) in P, the problems of finding a tour with a minimum number of turns that visits each point (edge) in S exactly once are also shown to be NP-hard. An approximation algorithm that finds a suboptimal path with the number of turns no greater than 3 times that of an optimal solution is also presented.

1 Introduction

In optimization problems, the objective is usually to seek an optimum value based on some predefined criteria and subject to a given set of constraints. One of the criteria often used in computational geometry is Euclidean distance, e.g., finding the shortest distance between two given points in the plane[10]. Recently, a new measure called the *link-distance*, has been proposed and studied first by Suri [18, 19]. Given two points x and y in the plane, and assuming the existence of some obstacles between them, the *link-length* of a path π connecting these two points, denoted by $|\pi|$, is simply the number of line segments in this path. The *link-distance* between x and y is the minimum link-length over all possible paths between x and y. This measure is of course meaningful only if the path is composed of straight line segments. Equivalently, the link-length and *link-distance* may be expressed in terms of the number of turns in a given path. The link-distance appears to be of great importance in robotics and communications systems where straight line motion or communication is relatively inexpensive but "*turns*" are costly. Consider for example Figure 1 in which a robot is to be launched from point x_1 to x_8. It is easy to show, using the triangle inequality, that the path $\pi = (x_1, x_2, \ldots, x_8)$ is shorter than $\pi' = (x_1, x_9, x_8)$. However, π contains six turns, while π' has only one turn which is more desirable if we take into account the considerable amount of time it will take the robot to make a single turn. More examples of the importance of this measure are found in the literature[14, 16, 17].

Many algorithms structured around the notion of link-distance have been devised to parallel those designed using the Euclidean distance. In [18], Suri gave a linear time algorithm for computing the shortest link-distance between two points inside a triangulated simple polygon. The link-center problem, which is defined as the locus of all points inside a simple polygon from which the link-distance to all other points is minimized, was solved by Lenhart et al in $O(n^2)$ time[11]. Recently, its time complexity was improved to $O(nlogn)$ by Djidjev *et al*[3] Also, some visibility problems can be regarded as link-path problems such as finding the region visible from a given point[4] or

[1]Department of Information and Computer Science, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia. e-mail: facp079@saupm00.bitnet.
[2]Supported in part by the National Science Foundation under Grant CCR 8901815. e-mail: dtlee@eecs.nwu.edu, Tel: 708-491-5007, Fax: 708-491-4455.

the set of points that are visible from all other points inside a given polygon[9], and the problem of finding the shortest line segment inside a simple polygon from which all the other points inside the polygon are visible[6]. On the other hand, some combinatorial problems, but not many, in computational geometry have been shown to be intractable, such as the art gallery problem which asks for the minimum number of positions required to *see* a simple polygonal area[8, 17] and the p-center problem which asks for the minimum number of unit circles that can cover n points in the plane[12].

In this paper, we investigate a variation of a class of problems pioneered by Chin and Ntafos[1, 2]. In [1, 2] ,they considered the problem of finding a route of minimum *Euclidean* length inside a polygon with the property that every point in the polygon is visible from at least one point along the route. They have shown that this problem is NP-hard for polygons with holes and for simple polyhedra using a simple reduction of the geometric traveling salesman problem to it. They have also presented an O(n log log n) algorithm for finding optimum routes in simple rectilinear polygons. For other related *watchman* route problems see [7, 15, 13]. In this paper, we consider the same optimization problem when the Euclidean metric is replaced with the link-distance measure.

Given a polygon P, call a line segment L inside P with the property that each point inside P is visible from at least one point on L a *visibility line segment*. Similarly, call a polygonal path π with the same property a *visibility path*. The *link-diameter* of P is defined to be the maximum link-distance between any two points inside P. It is obvious that if there exists a visibility line segment L inside P, then the link-diameter of P cannot exceed 3. This is because a path π with $|\pi| \leq 3$ between any two points x and y inside P can always be constructed by simply finding two other points x' and y' on L visible from x and y respectively, and then letting π be the path (x, x', y', y). Thus, if the link-diameter of P exceeds 3, then it is impossible to find a visibility line segment L inside P. In general, if the link-diameter of P is larger than $k + 2$ and $k \geq 1$, then a visibility path π with $|\pi| \leq k$ cannot exist. As we will show, for a given parameter k the problem of deciding if a visibility path π exists with $|\pi| \leq k$ is NP-hard even if the link diameter is known to be less than $k + 2$. In proving this main result, we show two other related problems to be intractable as well. These problems are summarized below:

Tour on a Set of Points. Given a polygonal area P and a set of points S on the boundary of P, no three of which are collinear, find a polygonal path inside P that visits each point in S exactly once and has a minimum number of turns.

Tour on a Set of Edges. Given a polygonal area P and a subset S of its edges, find a polygonal path inside P that visits each edge in S exactly once and has a minimum number of turns.

Visibility Path inside a Polygon. Given a polygonal area P, find a polygonal path π with a minimum number of turns inside P such that every point of P is visible from at least one point on π.

Throughout, we will work on the decision problems corresponding to these optimization problems. This is since a decision problem being NP-complete or NP-hard implies that its optimization version is NP-hard. This paper is organized as follows. Section 2 contains some preliminary definitions and conventions. Sections 3, 4 and 5 are devoted respectively to the proofs of NP-completeness or NP-hardness of the three aforementioned problems. In Section 6 an approximation algorithm is presented. Finally, Section 7 concludes with some discussion and remarks.

2 Definitions and Notations

A polygonal path π is a sequence of n points $(v_1, v_2, \dots, v_n)$, called *vertices*, joined by n line segments $\overline{v_1v_2}, \overline{v_2v_3}, \dots, \overline{v_{n-1}v_n}$, called *edges*. A polygonal path is called *simple* if no two nonconsecutive edges intersect. A polygon P of n vertices or n-gon is a polygonal path π of n vertices $(v_1, v_2, \dots, v_n)$, such that the last vertex v_n is connected to the first vertex v_1 by an edge $\overline{v_nv_1}$, and P is *simple* if no two nonconsecutive edges intersect. Throughout, we will be interested in simple polygons only, and hence the qualifier "*simple*" will be dropped. The boundary of P denotes the closed, and connected simple path defined by the sequence of the vertices. P will also denote the closed, finite, and connected region of the plane enclosed by the boundary of P. Given two points $x \in P, y \in P$, x is said to be *visible* from y, or x sees y if and only if the line segment joining them lies entirely within P. If S is a set of points in P, $x \in P$, we will define the function $V(x, S)$ to be the set of points in S visible from x, i.e., $V(x, S) = \{y | y \in S \text{ and } y \text{ is visible from } x\}$.

A *link path*, or simply a *path*, π between two points x_1 and x_k in a polygon P is a polygonal path $(x_1, x_2, \dots, x_k)$ inside P such that x_{i-1} and $x_i, 1 < i \le k$, are visible. Given such a path, we will call x_1 its *starting point*, x_k its *end point*, all other x_i's, $2 \le i \le k-1$, its *turns*.

Given a set of points S in P, a *tour* τ on S is a link path that visits each point in S exactly once, i.e., if $y \in S$, then y is a vertex in τ. A *subtour* τ' on S is simply a tour on S', where $S' \subset S$. If τ is a tour on S, τ' a subtour on S, then for convenience, we will make use of the set notation and write $\tau' \subseteq \tau$ to mean that τ visits all points visited by τ'. Also, if L is a line segment in τ, then we write $L \in \tau$.

Let $\tau = (x_1, x_2, \dots, x_k)$ be a tour defined on a set S. By definition, τ must visit all the points in S. It may happen that $\{x_1, x_2, \dots, x_k\} = S$, i.e., τ makes turns only at the points $x_2, \dots, x_{k-1}$. On the other hand, τ may make *extra* turns at points not in S. In this case, there is at least one vertex of τ that is not in S. For this purpose, we will define the *cost* of a tour τ, written $cost(\tau)$, to be the number of extra turns in τ, i.e., $cost(\tau) = j$ if and only if $|\tau| - |S| + 1 = j$. Finally, given a tour τ on a set S, τ will be called a *valid* tour if and only if $cost(\tau) = 0$, otherwise τ will be called *invalid*. Figure 2 shows an example of an invalid tour on $\{s, x_1, x_2, x_3, x_4, t\}$. The cost of this tour is 3 since there are three extra turns made at points y_1, y_2 and y_3.

3 Tour on a Set of Points

In this section we concern ourselves with the complexity of the problem of finding a tour on a set of points, abbreviated TSP. First, we consider a variant of TSP, which we will call TSP$'$ in which the tour on the set of points is restricted to start and end at two distinguished points $s, t \in S$. It is trivial to show that TSP$'$ can be solved in polynomial time using a nondeterministic program which in polynomial time can guess a tour and verify its cost. To show that TSP$'$ is NP-complete, we will reduce the *exact three cover* problem, abbreviated X3C, to it. Following the conventions in [5], we first rephrase TSP and TSP$'$ as decision problems and state the exact three cover problem.

Tour on a Set of Points (TSP)
INSTANCE: A polygon P with n vertices, a nonempty set S of points on the boundary of P no three of which are collinear, and an integer $c, 0 \le c \le n$. An instance of TSP will be represented by the 3-tuple (P, S, c).
QUESTION: Does there exist a tour τ inside P on S with the property that $cost(\tau) \le c$?

Restricted version of Tour on a Set of Points (TSP$'$)
INSTANCE: A polygon P with n vertices, a nonempty set S of points on the boundary of P no three of which are collinear, two points $s, t \in S$, and an integer $c, 0 \leq c \leq n$. An instance of TSP$'$ will be represented by the 5-tuple (P, S, s, t, c).
QUESTION: Does there exist a tour τ inside P on S that starts at s and ends at t with the property that $cost(\tau) \leq c$?

Exact Three Cover(X3C)
INSTANCE: A set $X = \{1, 2, \ldots, 3n\}$ and a collection $\mathcal{F} = \{F_1, F_2, \ldots, F_p\}$ of 3-element subsets of X. An instance of X3C will be represented by the 2-tuple $(X, \mathcal{F})$.
QUESTION: Does $\mathcal{F}$ contain an exact cover for X, i.e., a subcollection $\mathcal{F}' \subseteq \mathcal{F}$ such that every element of X occurs in exactly one member of $\mathcal{F}'$?

3.1 The reduction

Given an instance $I = (X, \mathcal{F})$ of X3C, we construct an instance $I' = (P, S, s, t, 0)$ of TSP$'$ such that the answer to I is **yes** if and only if the answer to I' is **yes**. In other words, in the constructed polygon, there is a valid tour on S if and only if X has an exact cover in $\mathcal{F}$. The constructed polygon will be composed of the following constructs (refer to the final construction in Figure 5). In these constructs, all joints and vertices of set elements constitute the set S.

1) The top middle joints (TMJ$'$s). Those are just points on the top boundary of the polygon, separated by notches so that no two of them see each other, i.e., they are pairwise invisible. There are $3n + 1$ top middle joints; the first $3n$, $\mathrm{TMJ}_1, \mathrm{TMJ}_2, \ldots, \mathrm{TMJ}_{3n}$ correspond respectively to the elements in $X = \{1, 2, \ldots, 3n\}$. The last one, TMJ_{3n+1} is added for the sake of the construction.

2) The top left joints (TLJ$'$s) and top right joints (TRJ$'$s). They are similar to the middle joints. Their total number is p, where $p = |\mathcal{F}|$. The right joints are numbered $\mathrm{TRJ}_1, \mathrm{TRJ}_3, \ldots$, i.e., odd-numbered, and the left joints are numbered $\mathrm{TLJ}_2, \mathrm{TLJ}_4, \ldots$, i.e., even-numbered. All the top joints combined, left, right and middle, are pairwise invisible from each other, except TMJ_{3n+1} and TRJ_1 since there is no notch between them.

3) The set elements (SE$'$s). This construct is the heart of the construction. If $F_t = \{i, j, k\} \in \mathcal{F}, 1 \leq t \leq p$, is an input set in the X3C, then there is one set element for each one of i, j and k. A set element that corresponds to the element i in set F_t will be denoted by $\mathrm{SE}_{t,i}$. Figure 3.a shows the construction of the set element $\mathrm{SE}_{t,i}$. In Figure 3.b, the vertex visibility graph $\mathrm{G}_{t,i}$ associated with $\mathrm{SE}_{t,i}$ is depicted. For each vertex in $\mathrm{SE}_{t,i}$ there is a corresponding node in $\mathrm{G}_{t,i}$, and there is an edge between two nodes in $\mathrm{G}_{t,i}$ if and only if their corresponding vertices in $\mathrm{SE}_{t,i}$ are visible from each other. The existence of a valid *subtour* inside a set element is tantamount to finding a Hamiltonian path that visits all nodes in its corresponding visibility graph. This implies that if a tour τ is to be valid, its part which visits the vertices in a set element must correspond to a Hamiltonian path in its corresponding visibility graph that starts at 1 and ends at 13 or starts at 5 and ends at 9. In other words, the following two conditions must be satisfied within *each* set element:

a) τ must either enter at 1 and exit at 13 or enter at 5 and exit at 9.

b) Once τ enters the figure of $\mathrm{SE}_{t,i}$, it must visit all the vertices before exiting. In other words, τ cannot enter, visit some of the vertices and come back later without making extra turns. For this reason, in what follows, we will write τ *visits* $\mathrm{SE}_{t,i}$ to mean "τ enters $\mathrm{SE}_{t,i}$, visits all the 13 vertices inside, and then exits."

4) The set constructs (SETC′s). In the final construction, there are p set constructs $\mathrm{SETC}_1, \mathrm{SETC}_2, \ldots, \mathrm{SETC}_p$ corresponding to the p sets $F_1, F_2, \ldots, F_p$ in the input to X3C. Figure 4 shows SETC_t, the set construct representation of $F_t = \{i, j, k\}$ with $i < j < k$. In this construct, three set elements for i, j and k, namely $\mathrm{SE}_{t,i}, \mathrm{SE}_{t,j}$ and $\mathrm{SE}_{t,k}$, are grouped together. Also, there are two joints: the left set joint, LSJ_t, and the right set joint, RSJ_t. This representation has the property that if a tour τ is to be valid, then either one of the following two situations must happen:

a) $\forall x \in \{i, j, k\}, \tau$ visits $\mathrm{TMJ}_x, \mathrm{SE}_{t,x}, \mathrm{TMJ}_{x+1}$ in this order.

b) τ visits the following in order: $\mathrm{LSJ}_t, \mathrm{SE}_{t,k}$, $\mathrm{SE}_{t,j}, \mathrm{SE}_{t,i}, \mathrm{RSJ}_t$.

Figure 5 shows the final construction. In the bottom, we concatenate p set constructs, one for each set in $\mathcal{F}$, the input to X3C. At the top, we draw the top joints, left, middle and right. The dotted lines in Figure 5 are visibility lines. The intention of this construction is the following (refer to Figure 5). We start at the point TMJ_1, go to one of the set element representations of 1, i.e., any $\mathrm{SE}_{t,1}$, then we go to TMJ_2, branch to one of the set elements of 2, i.e., any $\mathrm{SE}_{t,2}$, etc. Finally we get to TMJ_{3n+1}. We go from there to TRJ_1, then LSJ_1, the left set joint in SETC_1. At this point we have the following three cases:

1) If all the set elements in SETC_1 have been connected to top middle joints, then we just go directly to RSJ_1.

2) If none of the set elements in SETC_1, which corresponds to $F_1 = \{i, j, k\} \in \mathcal{F}$, have been connected to top middle joints, then we visit the following in order: $\mathrm{SE}_{1,k}$, $\mathrm{SE}_{1,j}, \mathrm{SE}_{1,i}, \mathrm{RSJ}_1$.

3) If some of the set elements have been connected to top middle joints but not all of them, then there is no possibility to visit them all without making extra turns. Consequently, a valid tour is impossible in this case. This happens if X does not have an exact cover in $\mathcal{F}$.

At this point we are at RSJ_1. From there we branch to the first top left joint, TLJ_2, and from there we go to the right joint in the second set construct, RSJ_2, and so on.

Figure 6 shows the construction of an instance of TSP′ from the X3C instance: X={1, 2, 3, 4, 5, 6} and $\mathcal{F}$={{1, 3, 4}, {1, 3, 5}, {2, 5, 6}}. In this figure, points TMJ_1 and RSJ_3 are the starting and end points respectively. Obviously, the input to X3C has an exact cover, and hence in the constructed polygon, there is a valid tour from TMJ_1 to RSJ_3. It can easily be checked that the following tour is valid: (TMJ_1, $\mathrm{SE}_{1,1}$, $\mathrm{TMJ}_2, \mathrm{SE}_{3,2}, \mathrm{TMJ}_3, \mathrm{SE}_{1,3}$, $\mathrm{TMJ}_4, \mathrm{SE}_{1,4}, \mathrm{TMJ}_5, \mathrm{SE}_{3,5}$, $\mathrm{TMJ}_6, \mathrm{SE}_{3,6}, \mathrm{TMJ}_7, \mathrm{TRJ}_1, \mathrm{LSJ}_1$, $\mathrm{RSJ}_1, \mathrm{TLJ}_2, \mathrm{RSJ}_2$, $\mathrm{SE}_{2,1}, \mathrm{SE}_{2,3}$, $\mathrm{SE}_{2,5}$, $\mathrm{LSJ}_2, \mathrm{TRJ}_3$, $\mathrm{LSJ}_3, \mathrm{RSJ}_3$).

3.2 Proof of NP-completeness

Lemma 1 *If τ is valid, then within each set element $\mathrm{SE}_{t,i}$, we must have*

a) $(1, 2, 3, 4, 5) \subseteq \tau$.

b) $(9, 10, 11, 12, 13) \subseteq \tau$.

c) $(6, 7, 8) \subseteq \tau$.

d) 1 *and* 13 *cannot be consecutive in* τ.

e) *Either* $(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13) \subseteq \tau$ *or* $(5, 4, 3, 2, 1, 8, 7, 6, 13, 12, 11, 10, 9) \subseteq \tau$.

Proof: The proof of this lemma follows directly from the necessity of the existence of a Hamiltonian path in each $\mathrm{G}_{t,i}$:

a) This is because $\deg(2) = \deg(4) = 2$, where $\deg(i)$ denotes the number of edges incident on vertex i and 3 is connected to both 2 and 4.

b) This is symmetrical to part **a**.

c) This is because $\deg(7) = 2$ with 6 and 8 being its neighbors.

d) If so, then by parts **a** and **b** $(9, 10, 11, 12, 13, 1, 2, 3, 4, 5) \subseteq \tau$. But part **c** above cannot be satisfied if τ is to be valid.

e) Applying parts **a, b, c** and **d** to the vertex visibility graph in Figure 3.b results in some of its edges being not qualified to be part of a Hamiltonian path. Figure 7 shows the remaining edges that may be part of the tour. If 1 is visited first, 6 must be visited after 5 for τ to be valid. Thus, by part **c** and symmetry, $(1, 2, ..., 13) \subseteq \tau$. On the other hand, if 5 is visited first, 8 must be visited after 1 and hence, by part **c** and symmetry again, $(5, 4, 3, 2, 1, 8, 7, 6, 13, 12, 11, 10, 9) \subseteq \tau$. □

Consider any set element in the final construction, say $\mathrm{SE}_{t,i}$. The only point in S outside $\mathrm{SE}_{t,i}$ visible from vertex 5 is TMJ_i. Similarly, the only point in S outside $\mathrm{SE}_{t,i}$ visible from vertex 9 is TMJ_{i+1}. Therefore, if the vertices within $\mathrm{SE}_{t,i}$ are visited in the order $5, \ldots, 9$,we must have $(\mathrm{TMJ}_i, 5, 4, 3, 2, 1, 8, 7, 6, 13, 12, 11, 10, 9, \mathrm{TMJ}_{i+1}) \subseteq \tau$. This corresponds in X3C to pairing $i \in X$ to $i \in F$ for some $F \in \mathcal{F}$. For convenience, we will say in this case that $\mathrm{SE}_{t,i}$ is "full with respect to τ" or just "full". If τ is valid, then the only other possibility, as stated in the lemma is that $\mathrm{SE}_{t,i}$ is entered at 1 and exited at 13, i.e., $(14, 1, 2, \ldots, 13, 15) \subseteq \tau$. In this case, we will say that $\mathrm{SE}_{t,i}$ is "empty with respect to τ" or just "empty". Also, we will say that a set construct SETC_t is "full with respect to τ" or just "full" if all of its set elements $\mathrm{SE}_{t,i}, \mathrm{SE}_{t,j}$ and $\mathrm{SE}_{t,k}$ are full, and "empty with respect to τ" or just "empty" if they are all empty.

Lemma 2 *If* τ *is valid, then*

a) $\forall i, 1 \le i \le 3n, \mathrm{TMJ}_i$ *is connected to* $\mathrm{SE}_{t,i}$ *for some* $t, 1 \le t \le p$.

b) $\forall t, 1 \le t \le p, \mathrm{SETC}_t$ *is either full with respect to* τ *or empty with respect to* τ.

Proof:

a) This follows from the fact that the first $3n$ middle joints can only see points inside the set elements. Specifically, TMJ_1 can only see vertex 5 inside some $\mathrm{SE}_{t,1}$ and $\mathrm{TMJ}_i, 2 \le i \le 3n$, can only see vertex 5 in $\mathrm{SE}_{u,i}$ and vertex 9 in $\mathrm{SE}_{v,i-1}$, where t, u and v are such that $1 \in F_t, i \in F_u$ and $i - 1 \in F_v$ in the input to X3C.

b) Let $\mathrm{SE}_{t,l}, \mathrm{SE}_{t,m}$ and $\mathrm{SE}_{t,r}$ be respectively the left, middle and right set elements inside SETC_t. It suffices to show that if one of them is empty, then the other two must both be empty. So, assume that the middle set element, $\mathrm{SE}_{t,m}$ is empty. Since $\mathrm{SE}_{t,m}$ is empty, it must be entered at vertex 1 and exited at vertex 13. But since outside $\mathrm{SE}_{t,m}$,the only point in S visible from

1 is vertex 13 of $\mathrm{SE}_{t,l}$, it must be the case that the last vertex visited in $\mathrm{SE}_{t,l}$ is 13, i.e., it is empty with respect to τ. Similarly, the first vertex visited in $\mathrm{SE}_{t,r}$ is 1, which means it must also be empty. This means if the middle set element is empty, both the other set elements must also be empty for τ to be a valid tour. The same reasoning applies if we start from the assumption that the left set element or the right set element is empty. It follows that either all the set elements inside SETC_t are full or empty, i.e., SETC_t is either full or empty. □

Lemma 3 *If τ is valid, then there are exactly n set constructs which are full with respect to τ.*

Proof: By Lemma 2.a, we must have $3n$ set elements which are full with respect to τ. By Lemma 2.b, each set construct is either full or empty. As a result, there are exactly $3n/3 = n$ full set constructs. □

Theorem 1 *TSP′ is NP-complete.*

Proof: TSP′ is in NP as we can nondeterministically guess a tour τ by giving the sequence of vertices of S visited by τ, and verify its cost in polynominal time. Given an instance $I = (X, \mathcal{F})$ of X3C, it is easy to construct an instance $I' = (P, S, s, t, 0)$ of TSP′ in polynomial time where $s = \mathrm{TMJ}_1$ and $t = \mathrm{LSJ}_p$ if p is even and RSJ_p if p is odd. It is important to note that the coordinates of the constructed polygon are rational numbers with polynomial size. This follows from the fact that no more than a constant number of additions and line intersections are needed to compute these coordinates. Now, we will show that the answer to I is **yes** if and only if the answer to I' is **yes**. In other words, in the constructed polygon, there is a valid tour if and only if X has an exact cover in $\mathcal{F}$. Suppose that the answer to I is **yes**. Then there is an exact cover, i.e., there are n 3-element subsets which cover the $3n$ elements. A valid tour in the constructed polygon can easily be constructed with exactly n full set constructs as explained in Section 3.1. On the other hand, suppose in the constructed polygon we have $cost(\tau) = 0$. Then, by Lemma 3, there are exactly n set constructs which are full with respect to τ. But each $\mathrm{TMJ}_i, 1 \leq i \leq 3n$, can be connected only to one of the set elements. It follows that the n full set constructs correspond to an exact cover for the instance I. □

Theorem 2 *TSP is NP-complete.*

Proof: As in the previous theorem, it is not difficult to see that TSP is in NP. Given an instance $I' = (P', S', s', t', c')$ of TSP′, we construct another instance $I = (P, S, c' + 2n)$ of TSP such that the answer to I' is **yes** if and only if the answer to I is **yes**. As shown in Figures 8.a and 8.b, P is identical to P' except for the addition of two polygonal narrow strips of n turns each, where n is the number of edges in P'. Also, two more points u and v are added, one at the end of each strip. Thus, we have $S = S' \cup \{u, v\}$. It is obvious that a tour on S with cost $\leq c' + 2n$ must have vertices u and v as its start and end points respectively, for otherwise the cost of the tour would be at least $3n$ in order to visit both u and v. Therefore the answer to I' is **yes**, i.e., there is a tour τ' inside P' on S' of length $\leq c'$ if and only if there is a tour $\tau = (u, s, \ldots, t, v)$ inside P on S of length $\leq c' + 2n$, i.e., the answer to I is **yes**. □

4 Tour on a Set of Edges

In this section we consider the second problem, TSE. First, we rephrase it as a decision problem and then we show that it is NP-hard by reducing TSP to it.

INSTANCE: A polygon P with n vertices, a nonempty subset $S = \{e_1, e_2, \ldots, e_k\}$ of edges of P and an integer $c, 0 \leq c \leq n$. An instance of TSE will be represented by the 3-tuple (P, S, c).
QUESTION: Does there exist a tour τ on the set of edges S inside P with the property that $cost(\tau) \leq c$?

The idea of the reduction is simple. We replace each point x to be visited in the input to TSP by an edge $e = \overline{xx'}$ in the input to TSE. This replacement is such that any tour on the set of edges can be forced to visit e at one of its endpoints, namely x, without increasing its length (note that it is sufficient to visit an edge at one of its endpoints). This is detailed in the proof of the following theorem.

Definition 1 *Let x be a point and $e = \overline{yy'}$ an edge on the boundary of a simple polygon such that x and y are mutually visible. We define the relation $Vis(x, e)$ as follows: if y is not the only point on e visible from x, then all points on e are visible from x. In other words, either y is the only point on e visible from x or all points on e are visible fron x. In the degenerate case in which e is a point, say y, $Vis(x, y)$ will simply mean that x and y are visible from each other.*

Theorem 3 *TSE is NP-hard.*

Proof: Given an instance $I = (P, S, c)$ of TSP, we construct another instance $I' = (P', S', c)$ of TSE such that for each point $x \in S$, there is a corresponding edge $e_x = \overline{xx'} \in S'$ called the *replacement edge* of x. The construction of P' is such that if e_x is the replacement edge of $x \in S$, then any tour that visits e_x can be forced to visit it at point x without increasing its length. It follows that if τ' is a tour on S' of length k, another tour τ on S of length k can easily be constructed by visiting the points in S in the order their replacement edges are visited in τ'. For simplicity, we will assume that no three points in S plus the vertices of P are collinear since the existence of such triplets will only complicate the proof. In order to find the replacement edge e_x for a point $x \in S$, we first find a *replacement line segment* L_x on the boundary of P such that for any two points x and y in S, $Vis(x, y)$ if and only if $Vis(x, L_y)$ and $Vis(y, L_x)$. The construction of replacement edges from replacement line segments is then accomplished by replacing each one of these line segments by a two-edge small outward notch. The calculation of the set of replacement line segments, which we will denote by $\mathcal{S}$, is carried out through the following three *refinement stages*.

a) Let $x \in S$ be a point on the boundary of P with its two adjacnt vertices u and v where u precedes v in a clockwise ordering of the vertices of P. If x is not a vertex of P, we will "pretend" that it is a vertex with $\angle uxv = \pi$. This stage simply assigns an initial replacemnt line segment $L_x = \overline{xx'}$, where x' is the midpoint of $\overline{vx}$.

b) In this stage, the replacement line segments calculated in stage **a** are "refined" by reducing their lengths, if necessary, to limit their visibility ranges so that at the end of this stage $\forall x, y \in S$ $Vis(x, y)$ if and only if $Vis(x, L_y)$ and $Vis(y, L_x)$. In order to achieve this, we consider each *ordered* pair L_x and $L_y (x \neq y)$ of replacement edges. That is, L_x is tested against L_y and L_y is tested against L_x in two different steps. Let L_x and $L_y (x \neq y)$ be the current (ordered) pair being considered. We have the following possibilities (refer to Figure 9).

- if $Vis(x, y)$ and x is the only point on L_x visible from y, then we leave L_x unchanged. In Figure 9.a, L_p is not changed after testing it against L_r and L_s.

- If $Vis(x, y)$ and x is not the only point on L_x visible from y, then we let z be a farthest point from x on L_x (i.e., between x and x') such that $Vis(y, \overline{xz})$ and let x' be the midpoint of $\overline{xz}$. In Figure 9.a, when testing L_s against L_t, we have $Vis(s, t)$ and s is not the only point on L_s visible from t. Therefore, the length of L_s is reduced so that it becomes all visible from point t as shown in Figure 9.b. Note that z (in this stage and the next stage below) always exists since we have assumed in the outset that no three points in S plus the vertices of P are collinear.
- If not $Vis(x, y)$, but part of L_x is visible from some point on L_y, then we let z be a farthest point from x on L_x such that no point on line segment $\overline{xz}$ is visible from any point on L_y and let x' be the midpoint of $\overline{xz}$. In Figure 9.b, when testing L_r against L_s, r and s are not visible from each other while L_r and L_s are weakly visible. Therefore, the length of L_r is reduced so that no point on it is visible from any point on L_s as shown in Figure 9.c.

It is not hard to see that at the end of this stage, the members of $\mathcal{S}$ must satisfy the following property:

Property 1 *Let $\overline{xx'}$ and $\overline{yy'}$ be two replacement line segments in $\mathcal{S}$. Then, $Vis(x, y)$ if and only if $Vis(x, L_y)$ and $Vis(y, L_x)$.*

Therefore, if $\overline{x''y''}$ is a line segment in a tour τ that visits $L_x = \overline{xx'}$ and $L_y = \overline{yy'}$ at points $x'' \in L_x$ and $y''\ inL_y$, where x'' is different from x, then it is always possible to replace x'' by x without increasing the length of τ.

c) At this point, there might exist two points p_x and p_y on two replacement line segments L_x and L_y respectively, such that the link-distance between p_x and p_y (that is greater than 1) is shorter than the link-distance betweeen x and y. For instance, in Figure 9.c, the link-distance between q and t is greater than the link distance between their respective line segments. In order to avoid this situation, we outline one more refinement stage which is basically the naive version of the algorithm of Suri to find the link distance between two points inside a simple polygon [18]. We do this refinement for each replacement line segment. So, assume we want to refine the position of point x', an end point of the replacement line segment L_x. For each $y \in S - x$, we do the following. We first find $V(y)$ which is the region visible from point y. We note that at this time, after stages **a** and **b** have been completed, either x is inside $V(y)$ or not. If it is, then we leave L_x unchanged. Otherwise there must exist a window w_1 that is an edge of $V(y)$ and intersects any path from y to x. We calculate $V(w_1)$ and continue, if necessary, finding the visibility polygons of the *unique* sequence of successive windows intersectig any path from y to x, and checking the inclusion of x and/or x' inside these visibility polygons and stoppping at the first window w_i for which the test of inclusion is successful. Once w_i is found, we proceed as in stage **b** to update x',if necessary, so that if $x \notin V(w_i)$ then $x' \notin V(w_i)$ .We ensure this condition by simply reducing the length of L_x if necessary. In Figure 9.c, $t' \in V(w)$ but t is not. Hence, t' is updated accordingly. This is done by letting z be the intersection of L_t and $V(w_i)$ and modifying t' to be the midpoint of $\overline{tz}$ (see Figure 9.d).

To finish the construction, we compute the replacement edges from these replacement line segments as follows. Let ϵ_v denote the smallest distance from any vertex v to the polygonal path

Π_v, which is the boundary of P with the two edges incident on v removed. Let $\epsilon = \min_{v \in P} \epsilon_v$. Consider *expanding* the boundary of P by a distance $\epsilon/3$. The resulting closed curve $\mathcal{P}$ consisting of straight line segments and circular-arcs is not self-intersecting. This curve will be used to calculate the lengths of replacement edges.

For each point $x \in S$, let $\mathcal{V}_x$ be the angular bisector of angle $\angle uxv$ that intersects $\mathcal{P}$ at point x''. Replace $L_x = \overline{xx'}$ by the two line segments, $\overline{x'x''}$ and $\overline{x''x}$. Here, $\overline{x''x}$ serves as the replacement edge of x.

The following property whose proof is a consequence of the construction of the edge $\overline{xx''}$ is important:

Property 2 *Let $\overline{xx''}$ and $\overline{yy''}$ be two replacement edges in P' of $x, y \in S$ obtained as described above. If p_x is any point on $\overline{xx''}$ and p_y is any point on $\overline{yy''}$, then the link distance between x and y is not greater than the link distance between p_x and P_y.*

It remains to show that a solution to I' is a solution to I. Obviously, if the answer to I' is **no**, then the answer to I is also **no**. So, let the answer to the instance $I' = (P', S', c)$ be **yes**. Then there exists a tour τ' on S' with $|\tau'| \leq c$. We construct another tour τ from τ'. τ visits the edges in S' in the order they are visited by τ'. The only difference between the two is that if $x \in S, e = \overline{xx''}$ and τ' visits e at a point z different from x, then we let τ visit e at x. **Property 2** guarantees that this is always possible.It follows that $|\tau| \leq |\tau'|$, i.e, τ is a tour on S with $|\tau| \leq c$. To finish the proof, we note that the transformation of P to P' requires no more than a polynomial number of finding the visibility polygons from a point or an edge plus some arithmetic operations and line intersections. □

5 Visibility Path inside a Polygon

In this section we consider the third problem, VPP. First, we prove that a special case of TSE which we will call TSE′ to be NP-hard. We have chosen to work on TSE′ because it is much easier to reduce it to our final problem, VPP. For convenience, let us call an edge connecting two vertices x and y *restricted* whenever one of x or y is convex in P. In TSE′, a set of restricted edges are to be connected together by polygonal paths.

TSE′:
INSTANCE: A polygon P with n vertices, a subset $S = \{e_1, e_2, \ldots, e_k\}$ of restricted edges of P and an integer $c, 0 \leq c \leq n$. An instance of TSE′ will be represented by the 3-tuple (P, S, c).
QUESTION: Does there exist c or less line segments that **connect** the set of restricted edges together? (see below).

VPP:
INSTANCE: A polygon P with n vertices and an integer $c, 0 \leq c \leq n$. An instance of VPP will be represented by the 2-tuple (P, c).
QUESTION: Does there exist a visibility polygonal path π inside P with $|\pi| \leq c$?

Note the major change in the objective of TSE′. Here, we are interested in finding the number of line segments that are used to *connect* those input edges of P. Two edges e_i and e_j are connected by a polygonal path π_{ij} if this path starts at one edge and ends at the other one. Moreover, if π_{jk} connects e_j and e_k, then π_{ij} and π_{jk} do not have to intersect edge e_j at the same point. Edges

$e_1, e_2, \ldots,$ and e_m are connected if, for $1 < j < m$, e_j is connected to both e_{j-1} and e_{j+1}. For example, In Figure 10.a, 7 line segments are used to connect the edges $e_1, e_2, \ldots, e_5$. Another important remark is the possibility for one line segment in a visibility path to be contained in another one. In Figure 10.b, the line segment $\overline{x_4x_5}$ is part of $\overline{x_3x_4}$. Consequently, the length of this path is 5; i.e., the vertex x_4 is a turn and *not* an endpoint. This is in accordance with the practical implication of a visibility path used by a watchman to guard a given polygonal area. In this example, the watchman travels the path in the order of its vertices; he goes from point x_3 to x_4 and then goes along the same path back to x_5.

Lemma 4 TSE$'$ *is NP-hard.*

Proof: The reduction from TSP to TSE is also a reduction from TSP to TSE$'$. □

5.1 The reduction

Given an instance $I = (P, S, c)$ of TSE$'$, we construct another instance $I' = (P', c + 2|S| - 2)$ of VPP such that the answer to I is **yes** if and only if the answer to I' is **yes**. This construction is detailed below. Before we describe the reduction, we need some definitions and notations. Let v be any reflex vertex adjacent to another vertex u in a polygon P. Let H be the infinite half line originating at u in the direction from u to v. Let x be the intersection of H and the boundary of P closest to v. Then the line segment $\overline{vx}$ is called a *window* w and $\overline{uv}$ the edge *generating* w. If w is a window generated by $\overline{uv}$, then it partitions P into two parts and that part which contains u will be called the *region of* w, denoted by $P(w)$. If w_1 and w_2 are two windows such that $P(w_1)$ is contained entirely within $P(w_2)$, then w_2 will be called *redundant*, otherwise it is *nonredundant*. The notion of redundant and nonredundant windows here is identical to that of the redundant and nonredundant chords defined in [6, 18, 19]. The following fact is a generalization of a theorem in [6] which says that a visibility line segment must intersect all the nonredundant windows in a given polygon.

Fact 1 *If π is a visibility path inside P, then π must intersect all the nonredundant windows in P.*

Given P, we construct P' such that the number of nonredundant windows in $P' = |S|$. Also, the construction is such that *part of* the visibility path inside P' which consists of c line segments serves as a solution to I. P' will be obtained from P by applying a number of transformations on P which are described in the following steps.

step 1. (refer to Figures 11.a and 11.b) Let $e \in S$ be one of the input edges in P. If u and v are its two endpoints then, since e is restricted by assumption, one of these two vertices, say v, is convex. Replace e by the polygonal path v, a, b, c, d, f, u where:

- **a)** The area defined by the simple polygon u, v, a, b, c, d, f lies entirely in the exterior of P except for points on edge e. This can be ensured by following a procedure similar to the one of replacing points by edges in the proof of Theorem 3.
- **b)** The points u, v, a, d, c are collinear.
- **c)** The lengths of the edges in this new construct is immaterial as long as the above two conditions are satisfied.

Let P_0 be the resulting polygon (Figure 11.b). Throughout, we will refer to the added construct v, a, b, c, d, f, u in place of e as *subpolygon(e)*. In *subpolygon*(e) there is a nonredundant window generated by edge $\overline{cd}$ connecting a and d. We will refer to this window as *window(e)*. For example, in Figure 11.b, $window(\overline{uv}) = w = \overline{ad}$ and $window(\overline{u'v'}) = w' = \overline{a'd'}$. For brevity, we will call members of $\{window(e)|e \in S\}$ *desirable* windows, and all other windows *undesirable*. An important observation is that all desirable windows are nonredundant. Let W be the set of nonredundant windows in P_0. Obviously, $|S| \leq |W|$. Note also that e in P becomes a *redundant* window in P_0.

step 2. This step consists of a series of transformations after the end of which, we obtain another polygon P' having the property that its set of nonredundant edges is exactly S, the set of desirable windows. First, we transform P_0 to P_1, then we transform P_1 to P_2 and so on until we get to $P_k = P'$ on which no more transformations can be done. In each iteration, we choose a nonredundant window that is undesirable and make it redundant by removing some part of the polygon. Let $w = \overline{vx}$ be a nonredundant but undesirable window in P generated by the edge $\overline{uv}$ which is adjacent to the edge $\overline{vy}$ (see Figure 12.a). Let also T be the set of all vertices together with all nonredundant windows' endpoints on the boundary of $P_i - P(W)$ except x and v, the endpoints of w. Find a set $T' \subseteq T$ of points visible from v and sort them in ascending order of their polar angles around v with respect to w in the direction from x to v. Find $x' \in T'$(if any) such that $\angle xvx'$ is minimum (break ties by choosing x' to be closest to v) and such that one of the following two conditions holds:

- **a)** One desirable window becomes entirely inside the subpolygon defined by the closed polygonal path $v, u, \ldots, x, \ldots, x', v$. Note that there were no desirable windows inside this region, for otherwise w would have not been nonredundant.
- **b)** The direction of $\overline{vx'}$ coincides with $\overline{vy}$, the edge adjacent to $\overline{uv}$. In this case, y is reflex, for otherwise, case a would have been satisfied by another point x'' with $\angle xvx'' < \angle xvx'$.

Let u' be a point on the boundary of $P(w)$ that is closest to v on the extension of $\overline{x'v}$ in the direction from x' to v. Now, remove the region defined by the polygonal path $(v, u, ..., u')$, which is a portion of the boundary of $P(w)$, and the line segment $\overline{vu'}$. Finally, update the set of nonredundant windows. Figures 12 and 13 show examples of these two cases. In the first case (Figure 12.b), the nonredundant window w is transformed into a redundant window, $w' = \overline{vx'}$. In the second case w is removed, and the number of vertices in P_i is reduced. Repeat this process as long as there is a nonredundant window which is undesirable.

5.2 Proof of NP-hardness

It is important to note that, if w is a nonredundant window, then it is always possible to find a point x' as defined in step 2 above so that either case a or b must occur in each iteration. Furthermore, the series of transformations must terminate as the number of vertices and/or nonredundant windows of the polygon is reduced by at least one in each iteration. This guarantees that the reduction always results in a new polygon P' with no nonredundant windows except the desirable ones. Thus, we have the following observation:

Observation 1 *If* $I' = (P', c + 2|S| - 2)$ *is an instance of* VPP *constructed from the instance* $I = (P, S, c)$ *of* TSE$'$ *as described above, then the set of nonredundant windows in* P' *is exactly* $\{window(e)|e \in S\}$.

Lemma 5 *For $0 \leq i \leq k-1$, let P_i be transformed into P_{i+1} as described in step 2 of the reduction. Then there exists a path inside P_i of length $\leq c$ that visits each and every desirable window of P_i if and only if there exists a path inside P_{i+1} of length $\leq c$ that visits each and every desirable window of P_{i+1}.*

Proof: We first prove the lemma for case a in step 2 of the reduction. Consider Figure 14.a in which u', u, v, x, x' and the windows w and w' are shown. Let π be a path inside P_i of length $\leq c$ that visits each and every desirable window of P_i. First, we may assume that π is optimal, i.e., it consists of a minimum number of line segments. We show here that region R defined by the boundary of P_i between v and u' and the line segment $\overline{vu'}$ need not be visited by π. Note that π starts and ends inside $P_i - P(w)$ which contains all the desirable windows. As a result, if π crosses w at all, then the number of its intersection points with w is even (see Figure 14.a). Let us assume also that π makes only two intersection points with w since working with more than two is just a reiteration of the proof.

Suppose that region R must be visited in order to achieve optimality. Then π must visit R at some point,say s. Let the two turns before and after s be p and q respectively. Obviously, p and q are not visible from each other and therefore there must exist a (reflex) vertex, say t, on the boundary of P_i between x and x'. Moreover, the portion of P_i defined by the closed polygonal path $v, u, \ldots, x, \ldots, t, v$ must contain a desirable window (see Figure 14.b). But $\angle xvt < \angle xvx'$. This means that t should have been chosen in step 2, not x'.

As to case b, there are no desirable windows to the left of w' and hence, by a similar argument, region R need not be visited by π.□

Lemma 6 *The elements of S can be connected by c line segments or less if and only if there exists a visibility path π inside P' with $|\pi| \leq c + 2|S| - 2$.*

Proof: Let $I = (P, S, c)$ be an instance of TSE$'$ and $I' = (P', c + 2|S| - 2)$ an instance of VPP where P' is obtained from P using the reduction described in the previous section. Let W be the set of nonredundant windows in P' and $N = \{e | e \in S\}$, a set of some of the redundant windows in P'. By **Fact**1 above, $W = \{window(e) | e \in N\}$ and any path inside P' is a visibility path if and only if it visits all of the windows in W.

Suppose that π is a visibility path inside P' such that $|\pi| \leq c + 2|S| - 2$. Suppose also that π intersects one of the elements of N, say $\overline{uv}$, at points that are not vertices (turns) of π (see Figure 15.a). π must make *exactly* two turns inside $subpolygon(\overline{uv})$, x and z in this Figure. Intuitively, π can easily be changed in this region such that its two turns are forced to lie on e (x' and z' in Figure 15.b) without increasing its length (recall that, according to the definition of a visibility path, y in Figure 15.b is a turn and not an endpoint of π). The same reasoning applies if π intersects $e \in N$ at only one point; this happens if y in Figure 15.a is an endpoint of π. Thus, we may assume without loss of generality that π intersects all windows in N at its vertices only. Now, it is easy to pick c line segments in π that serve to connect all edges in S, i.e., a solution to I. This is simply achieved by removing all portions of π that lie inside the $|N|$ regions defined by $\{subpolygon(e) | e \in N\}$. Figure 15.c shows the region $subpolygon(\overline{uv})$ shown in Figure 15.b after removing the two line segments $\overline{x'y}$ and $\overline{z'y}$. Note that in this case, $2|N| - 2 = 2|S| - 2$ line segments are removed from π (π starts and ends at two of the nonredundant windows in W).

On the other hand, if it is possible to connect all the elements of S inside P using $\leq c$ line segments, then reversing the outlined procedure above in the obvious way, we can add exactly $2|N| - 2 = 2|S| - 2$ line segments inside P' in order to build a visibility path of length $\leq c + 2|S| - 2$.
□

Lemma 7 *The construction of P' from P is achievable in polynomial time.*

Proof:

First, we show that the number of iterations in step 2 of the reduction is $O(n)$. As pointed out earlier, after each iteration, the number of vertices is reduced by at least one if case b happens. If case a happens, then a nonredundant window is transformed into a redundant one and it will stay redundant for the rest of the transformation since its region will contain a desirable window. However, in this case, a redundant window may become nonredundant. Thus, the number of transformations in step 2 is bounded above by the number of nonredundant (that are undesirable) windows plus the redundant ones which is O(n). To finish the proof, we note here as in the proofs of Theorems 1 and 3, assuming that the coordinates of the input polygon are rational numbers, that the calculations of the constructed polygon's vertices are done with a constant number of line intersections and additions, so they are rational numbers with polynomial size. Since the construction time is polynomial, the entire transformation is achieved in polynomial time.□

Theorem 4 *VPP is NP-hard.*

Proof: Given an instance I of TSE$'$, we have shown how to transform it into another instance I' of VPP above. Lemma 4 shows that TSE$'$ is NP-hard, Lemma 7 shows that this transformation can be done in polynomial time, and Lemmas 5 and 6 establish its correctness. □

6 Finding an approximate solution for the visibility path problem

In the following, we describe a simple heuristic to find an approximate visibility path inside a given polygon P. The heart of this heuristic is based on a well known algorithm to find an approximate solution of a given instance of the traveling salesman problem and on the idea of visibility windows introduced by Suri [18, 19]. The first step is to compute all the set of windows which takes $O(n \log n)$ time in the worst case. Figure 16 shows a polygon P with its set of windows. The next step is to remove all the redundant windows and keep only the nonredundant ones in linear time [18, 19]. Figure 17 shows only the nonredundant windows.

Let $w_1, w_2, \ldots, w_k$ be k nonredundant windows with the property that $P(w_1) \cap P(w_2) \cap \ldots \cap P(w_k) = Q$ is not empty. If $k = 1$ (e.g. w_6 in Figure 17) then remove Q from P to get another polygon $P' = P - Q$. Otherwise, Q is a subpolygon of P of the form $v_i, v_{i+1}, \ldots, v_j, u_1, u_2, \ldots, u_{k-1}$, where $v_i, v_{i+1}, \ldots, v_j$ is part of the boundary of P and $u_1, u_2, \ldots, u_{k-1}$ is an inward convex polygonal path inside P defined by the intersections of $w_1, w_2, \ldots, w_k$. In Figure 17 above, Q is the subpolygon $v_1, v_2, v_3, v_4, v_5, u_1, u_2$. It can be shown that v_i and v_j are visible from each other, or in other words, the line segment connecting v_i and v_j lies entirely inside P. Now, replace the polygonal path $v_i, v_{i+1}, \ldots, v_j$, which is part of the polygon boundary by a new edge $\overline{v_i v_j}$ in P'. If we repeat this process until all the nonredundant windows are exhausted, then we have as a result a polygon, say P'', in which each edge is either an edge in P, a nonredundant window in P or an edge introduced by removing a convex polygonal path as described above. In the latter two cases, we will call such edges *special* edges. Finding all the special edges can be accomplished in $O(n^2)$ time. Figure 18 shows P'' which resulted from applying this process to the polygon in Figure 17. Those special edges which were nonredundant windows must be visited by any visibility path, while it is sufficient, but not necessary, to visit those induced by the convex polygonal paths. Thus, if a path visits all the edges, then it is a visibility path. Consequently, finding a visibility path reduces to finding a polygonal path inside P'' that intersects all the special edges in P''. We proceed to find such a path as follows:

1. Find $\Pi(e_i, e_j)$, a polygonal path of minimum link-length between each pair of special edges e_i and e_j in P''.

2. Build a complete graph $G = (V, E, C)$ with:
V = the set of special edges e_i in P''.
$E = \{d_{i,j} | e_i$ and e_j are special edges in $P''\}$ and
$C = \{c_{i,j}$ = link-length of $\Pi(e_i, e_j)$ = link-distance between e_i and e_j $\}$.

3. Find a minimum spanning tree T of G and convert it into a directed graph D by replacing each edge by a two-edge directed cycle. D is an Eulerian circuit.

4. Starting from a leaf node e_{i_0} of T, traverse the edges of D so as to convert the directed circuit into a path $(e_{i_0}, e_{i_1}, \ldots, e_{i_k})$ skipping nodes that have been visited before. Let $Seq(e_{i_0}, e_{i_k})$ denote the sequence of edges resulting from concatenating the polygonal paths $\Pi(e_{i_0}, e_{i_1})$, $\Pi(e_{i_1}, e_{i_2})$, $\ldots$, $\Pi(e_{i_{k-1}}, e_{i_k})$.

5. If for some $j, \Pi(e_{i_{j-1}}, e_{i_j})$ and $\Pi(e_{i_j}, e_{i_{j+1}})$ are two polygonal paths in $Seq(e_{i_0}, e_{i_k})$ that intersect the special edge e_{i_j} at two different points x and y, then insert the line segment $\overline{xy}$ into $Seq(e_{i_0}, e_{i_k})$ between $\Pi(e_{i_{j-1}}, e_{i_j})$ and $\Pi(e_{i_j}, e_{i_{j+1}})$. So, this portion of $Seq(e_{i_0}, e_{i_k})$ is updated to be $(\ldots, \Pi(e_{i_{j-1}}, e_{i_j}), \overline{xy}, \Pi(e_{i_j}, e_{i_{j+1}}), \ldots)$.

Let π, the approximate visibility polygonal path, be $Seq(e_{i_0}, e_{i_k})$ resulting from step 5. Since the link-distance measure obeys the triangle inequality, the total link-length of T obtained in step 3 cannot exceed the link-length of an optimal visibility path. Thus, the path resulting from step 4 cannot exceed twice the link-length of an optimal visibility path. The number of line segments, if any, that are added in step 5 is at most the number of special edges to be visited minus two, which is at most the link-length of an optimal path minus 3. Consequently, the link-length of π is strictly less than 3 times the link-length of an optimal path. This performance ratio can be reduced to 2.5 by simply employing the modified heuristic of the Traveling Salesman Problem in [5] which has the performance ratio 1.5. The time complexity of this heuristic is dominated by step 1, computing the $\Pi(e_{i,j})'s$, which takes $O(n^3 \log n)$ time.

Given a simple polygon P, let OPT(P) denote the length of an optimal visibility path inside P and APPROX(P) the length of an approximate visibility path obtained using the heuristic procedure outlined above. We have in essence proved the following theorem.

Theorem 5 *For all simple polygons P with n vertices, a suboptimal visibility path π can be obtained in $O(n^3 \log n)$ time such that its link length* APPROX(P) $< 3\times$ OPT(P).

7 Conclusion

We have shown that the problem of finding a polygonal path of minimum number of turns inside a given simple polygon such that the entire polygon is visible from at least one point on the path is NP-hard. As a by-product, we have also shown that given n points (edges) on the boundary of a simple polygon, the problem of finding a tour that visits each and every point (edge) exactly once such that the tour has a minimum number of links is NP-complete (NP-hard). We have also presented an approximation algorithm that finds a suboptimal path with the number of turns no greater than 3 times that of the optimal solution in $O(n^3 \log n)$ time, where n is the number of vertices of the given polygon. Although the problem of deciding if there exists a polygonal path with k turns, for arbitrary k is NP-hard, the problem of deciding if such a path exists for a fixed k remains open. Furthermore, we pose the problem of identifing a class of polygons for which a visibility line segment ($k = 1$) exists. A characterization would generalize the class of polygons, known as *star-shaped* polygons, for which a *visibility point* or *kernel* exists.

Acknowledgement
The authors are grateful to the anonymous reviewers for their valuable comments that have helped improve the quality of the presentation of this paper.

References

[1] W. P. Chin and S. Ntafos, "Optimum watchman routes," *Information Proc. Letters* 28, 1988, 39-44.

[2] W. P. Chin and S. Ntafos, "Watchman routes in simple polygons," *Discrete and Computational Geometry*, 6,1, 1991, 9-31.

[3] H. Djidjev, A. Lingas, and J.-R. Sack, "An $O(n \log n)$ algorithm for computing the link center of a simple polygon," *Discrete & Comput Geometry* 8, 1992, 131-152.

[4] H. ElGindy and D. Avis, "A linear algorithm for computing the visibility from a point," *Journal of Algorithms*, 2, 1981, 186-197.

[5] R.M. Garey and D.S. Johnson, Computers and Intractability, W.H. Freeman and Company, New York, 1979.

[6] Y. Ke, "Detecting the weak visibility of a simple polygon and related problems," Technical report, Johns Hopkins University, 1987.

[7] E. Kranakis, D. Krizanc and L. Meertens, "Link length of rectilinear watchman tours in grids," *Proc. 2nd Canadian Conf. Comput. Geom.*, 1990, 328-331.

[8] D.T. Lee and A. K. Lin, "Computational Complexity of Art Gallery Problems," *IEEE Trans. Information Theory* IT-32, 1986, 276-282.

[9] D.T. Lee and F. P. Preparata, "An optimal algorithm for finding the kernel of a polygon" *J. of the ACM* 26, July 1979, 415-421.

[10] D.T. Lee and F.P. Preparata, "Euclidean shortest paths in the presence of rectilinear barriers," *Networks* 14(3), 1984, 393-410.

[11] W. Lenhart, R. Pollack, J. Sack, R. Seidel, M. Sharir, S. Suri, G.T. Toussaint, S. Whitesides and C. Yap, "Computing the link center of a simple polygon," *Discrete & Computational Geometry*, 3, 1988, 281-293.

[12] N. Megiddo and K. Supowit, "On the complexity of some common geometric problems," *SIAM J. Comput.*, vol. 13, No. 1, Feb. 1984, 182-196.

[13] J. S. B. Mitchell and E.L. Wynters, "Watchman routes for multiple guards," Manuscript, Cornell Univ., School of Oper. Res. and Indust. Engrg., 1991.

[14] W. Niedringhaus, “Scheduling without queuing: The space factory problem,” Technical report, Princeton University, 1979.

[15] S. Ntafos, “Watchman routes under limited visibility,” *Proc. 2nd Canadian Conf. Comput. Geom.*, 1990, 89-92.

[16] J. O’Rourke, “The complexity of computing minimum convex covers for polygons”, *Proc. 20th Allerton Conf.*, 1982, pp. 75-84.

[17] J. O’Rourke, Art Gallery Theorems and Algorithms, Oxford University Press, Oxford, 1987.

[18] S. Suri, “A linear time algorithm for minimum link-paths inside a simple polygon,” *Computer Vision, Graphics and Image Processing* 35, 1986, 99-110.

[19] S. Suri, “Minimum link paths in polygons and related problems,” PhD thesis, Computer Science Department, John Hopkins University, August 1987.