Available online at www.sciencedirect.com

SCIENCE DIRECT®

Electronic Notes in Theoretical Computer Science

Electronic Notes in Theoretical Computer Science 120 (2005) 45–57

www.elsevier.com/locate/entcs

On the Complexity of Finding Paths in a Two-Dimensional Domain II: Piecewise Straight-Line Paths

Arthur W. Chou[1]

Department of Mathematics and Computer Science
Clark University
Worcester, MA 01610, USA

Ker-I Ko[2]

Department of Computer Science
State University of New York at Stony Brook
Stony Brook, NY 11794, USA

Abstract

The problem of finding a piecewise straight-line path, with a constant number of line segments, in a two-dimensional domain is studied in the Turing machine-based computational model and in the discrete complexity theory. It is proved that, for polynomial-time recognizable domains associated with polynomial-time computable distance functions, the complexity of this problem is equivalent to a discrete problem which is complete for Σ_2^P, the second level of the polynomial-time hierarchy.

Keywords: computational complexity, piecewise straight-line path, polynomial-time hierarchy, two-dimensional domain

1 Introduction

Finding a path in a two-dimensional region is an important problem in computational complex analysis [7], computational geometry [5,6,12], and robotics

[1] Email: achou@black.clarku.edu

[2] Email: keriko@cs.sunysb.edu

doi:10.1016/j.entcs.2004.06.033

[8]. In Chou and Ko [2], we studied the computational complexity of finding the shortest path that connects two given points and lies entirely in a given two-dimensional domain. In this paper, we continue this investigation to study the computational complexity of finding a piecewise straight-line path that connects two given points with a constant number of straight-line segments which lie entirely in a given domain (called the *k-path problem* if the number of line segments in the path is required to be at most k). This problem has been studied in the context of computational geometry [16], robotics [14], and VLSI layout theory [15,17].

As in Chou and Ko [2], we study the k-path problem in the context of computable analysis [13,18] and complexity theory of real functions of Ko and Friedman [11,9,10]. The basic framework for this study can be summarized as follows (for the technical details, see Section 2 and [2]):

(1) The basic computational model for real-valued functions is the oracle Turing machine, to which the real number input is presented in the form of an oracle function mapping integers to dyadic rationals.

(2) We consider only domains with polynomial-time representations. That is, we ask what the computational complexity of the k-path problem is, if the given domain S has a polynomial-time representation.

(3) We consider two types of domains with polynomial-time representations: (A) Bounded, simply connected domains with polynomial-time computable boundary curves; and (B) Polynomial-time recognizable domains with polynomial-time computable distance functions.

(4) The oracle Turing machines that solve the k-path problem are allowed to make mistakes, with errors occur only when the target path is very close to the boundary of the given domain S.

In the following, we say that the k-path problem (or the shortest path problem) with respect to domain S is *$\mathcal{C}$-hard* for some discrete complexity class $\mathcal{C}$, if every problem $B \in \mathcal{C}$ can be solved in polynomial time relative to the solution of the k-path problem (or, respectively, the shortest path problem) of domain S. Based on this notion of hardness, we can summarize the main results of Chou and Ko [2] as follows:

(I) If a domain S satisfies property (A) or property (B) defined above, then the corresponding shortest path problem is solvable by an oracle Turing machine in polynomial space.

(II) There exists a domain S satisfying property (A) such that the corresponding shortest path problem is $\#P$-hard.

(III) There exists a domain S satisfying property (B) such that the corresponding shortest path problem is *PSPACE*-hard.

In addition to these results about the shortest path problem, Chou and Ko [2] also proved the following upper bound result about the 1-path problem (called the *straight-line path problem* in [2]):

(IV) If a domain S satisfies property (A) or property (B), then the corresponding 1-path problem is solvable by a Π_1^P oracle Turing machine (i.e., the complement of the 1-path problem is solvable by a polynomial-time nondeterministic oracle Turing machine).

Regarding the computational complexity of the k-path problem, we prove, in this paper, the following results:

(V) There exists a domain S satisfying properties (A) and (B) such that the corresponding 1-path problem is Π_1^P-hard.

(VI) For any $k \geq 2$, if a domain S satisfies property (A) or property (B), then the corrsponding k-path problem is solvable by a Σ_2^P oracle Turing machine (i.e., it is solvable by a polynomial-time nondeterministic oracle Turing machine relative to an oracle set in NP).

(VII) For any $k \geq 2$, there exists a domain S satisfying property (B) such that the corresponding k-path problem is Σ_2^P-hard.

Since the basic notations and computational models are similar to those used in [2], we will only present, in Section 2, the formal model for computing k-path problem. The reader is referred to [2] for the motivation and properties of this computational model. The main results (V)–(VII) are then presented in Sections 3 and 4. Section 5 discusses the open questions about the k-path problem.

2 Computational Model for the k-Path Problem

We will characterize the computational complexity of the k-path problems in terms of discrete complexity classes in the polynomial-time hierarchy. These complexity classes include P, NP, Π_1^P ($coNP$), Σ_2^P.

Sets in the polynomial-time hierarchy have simple bounded-quantifier characterizations (see, e.g., Du and Ko [4]). In particular, a set $A \in \{0,1\}^*$ is in Σ_2^P if there exist a set $B \in P$ and a polynomial function p such that, for any $w \in \{0,1\}^*$ of length n,

$$w \in A \iff (\exists u, |u| = p(n))\,(\forall v, |v| = p(n))\ \langle w, u, v\rangle \in B.$$

For the computational model of real functions, we follow the basic approach of the Turing machine-based complexity theory of real functions (see [9,10]). Let $\mathbb{D}$ denote the set of dyadic rationals; that is, $\mathbb{D} = \{m/2^n : m \in \mathbb{Z}, n \in \mathbb{N}\}$.

For any $n \geq 0$, let $D_n = \{m/2^n : m \in \mathbb{Z}\}$. We say a function $\phi : \mathbb{N} \to \mathbb{D}$ *binary converges* to a real number x, or *represents a real number* x, if (i) for all $n \geq 0$, $\phi(n) \in \mathbb{D}_n$, and (ii) for all $n \geq 0$, $|x - \phi(n)| \leq 2^{-n}$.

The basic model for the computation of real functions is the oracle Turing machine, which uses a function ϕ as an oracle and takes an integer $n > 0$ as the input. Intuitively, the oracle ϕ represents a real number x and the input n denotes the output precision.

Definition 2.1 (a) A function $f : \mathbb{R} \to \mathbb{R}$ is said to be *computable* if there is an oracle Turing machine M that, on an oracle function $\phi : \mathbb{N} \to \mathbb{D}$ that binary converges to a real number x and an input $n \in \mathbb{N}$, outputs a string $d \in \mathbb{D}_n$ such that $|d - f(x)| \leq 2^{-n}$.

(b) A function $f : [0, 1] \to \mathbb{R}$ is *polynomial-time computable* if it is computable by an oracle Turing machine M that operates in polynomial time (i.e., for any oracle function ϕ, the machine M always halts on input n in time $p(n)$ for some polynomial function p).

The above definition can be extended to functions defined on two dimensional plane in a natural way. In particular, an oracle Turing machine computing a function $f : \mathbb{R}^2 \to \mathbb{R}^2$ uses two oracles that represent a point $\mathbf{z}$ in $\mathbb{R}^2$, and it outputs a pair $\langle d, e\rangle$ of dyadic rationals as an approximation to the output point $f(\mathbf{z})$ in $\mathbb{R}^2$.

We are interested in the following path-finding problem on the two dimensional plane $\mathbb{R}^2$:

k-Path Problem ($k \geq 1$): Let S be a fixed domain in the two-dimensional plane $\mathbb{R}^2$. Given two points $\mathbf{x}, \mathbf{y}$ in S, determine whether there is a path π from $\mathbf{x}$ to $\mathbf{y}$ which consists of at most k straight-line segments in domain S (such a path is called a *k-segment path*).

Following the approach of Chou and Ko [2], we will study this problem with respect to domains S which have the following types of *polynomial-time representations*:

- We say S has *Property* (A) if S is bounded, simply connected (i.e., connected and having no holes), and its boundary Γ_S is a polynomial-time computable Jordan curve (i.e., it is the image of a one-to-one, polynomial-time computable function $f : [0, 1] \to \mathbb{R}^2$, except that $f(0) = f(1)$).
- We say S has *Property* (B) if S is bounded, connected, polynomial-time recognizable, and the function $\delta_S(\mathbf{x})$ is polynomial-time computable.

In the above, the function $\delta_S(\mathbf{x})$ denotes the Euclidean distance between a point $\mathbf{x} \in \mathbb{R}^2$ and the boundary Γ_S of domain S; that is, $\delta_S(\mathbf{x}) =_{\text{defn}}$

$\text{dist}(\mathbf{x}, \Gamma_S) = \min\{|\mathbf{x} - \mathbf{y}| : \mathbf{y} \in \Gamma_S\}$. (In general, for two closed sets A, B, we define $\text{dist}(A, B) = \min\{|\mathbf{x} - \mathbf{y}| : \mathbf{x} \in A, \mathbf{y} \in B\}$.) Also, a set $S \subseteq \mathbb{R}^2$ is called *polynomial-time recognizable* if there exist an oracle Turing machine M and a polynomial p such that $M^{\phi,\psi}(n)$ computes the characteristic function $\chi_S(\mathbf{z})$ of domain S in time $p(n)$ whenever (ϕ, ψ) represents a point $\mathbf{z}$ in $\mathbb{R}^2$ whose distance to the boundary Γ_S of S is greater than 2^{-n}; i.e., the set

$$E_n(M) = \{\mathbf{z} \in \mathbb{R}^2 : (\exists\, (\phi, \psi) \text{ representing } \mathbf{z})\ [M^{\phi,\psi}(n) \neq \chi_S(\mathbf{z})]\}$$

is a subset of $\{\mathbf{z} \in \mathbb{R}^2 : \delta_S(\mathbf{z}) \leq 2^{-n}\}$. For more discussion about the notion of polynomial-time recognizable sets and the distance functions, see [1,2,3].

With respect to these notions of polynomial-time computability of a domain, the question we are concerned with is the following: Suppose a domain S has property (A) or property (B). What is the complexity of the k-path problem of domain S?

Similar to the shortest path problem, if we regard the k-path problem of a domain S as the characteristic function f_S of domain S, which maps a pair of points $\mathbf{x}, \mathbf{y} \in \mathbb{R}^2$ to $\{0, 1\}$ (with 0 denoting NO and 1 denoting YES), then f_S is not a continuous function, and hence is not computable (as it is well known in computable analysis that a computable real-valued function must be continuous). Therefore, the natural model of computation is to allow the oracle Turing machine that computes the function f_S to make mistakes. The following definition is similar to that about polynomial-time solvable shortest path problem given in [2].

Definition 2.2 Let $k \geq 1$.

(a) We say the k-path problem with respect to a domain S is *computable* if there exists an oracle Turing machine M, that uses four oracles ϕ_{x_1}, ϕ_{x_2}, ϕ_{y_1}, ϕ_{y_2} representing two points $\mathbf{x}, \mathbf{y} \in S$, and takes an integer $n \in \mathbb{N}$ as the input, such that the following conditions hold:

- (i) M always halts and outputs either 0 or 1.
- (ii) If there is a k-segment path π in S between $\mathbf{x}$ and $\mathbf{y}$ such that $\text{dist}(\pi, \Gamma_S) \geq 2^{-n}$, then M outputs 1.
- (iii) If $\delta_S(\mathbf{x}) \geq 2^{-n}$, $\delta_S(\mathbf{y}) \geq 2^{-n}$, and there is no k-segment path π in S between $\mathbf{x}$ and $\mathbf{y}$, then M outputs 0.

(b) We say the k-path problem with respect to a domain S is *polynomial-time computable* if it is computed by an oracle Turing machine M as described in (a) above, that satisfies the following extra condition:

- (iv) There is a polynomial function p such that, for any oracles, M on input n halts in $p(n)$ steps.

3 Upper Bounds

We first consider the case of $k = 1$. The following lemma regarding the 1-path problem has been proven in [2]:

Lemma 3.1 *Assume that S has property (A) or property (B). Then, there is a polynomial-time nondeterministic oracle Turing machine M such that, for any given oracles $\phi_1, \phi_2, \phi_3, \phi_4$ representing two points $\mathbf{x}, \mathbf{y} \in S$, respectively, and for any input $n \in \mathbb{N}$, the following conditions hold:*

(a) If the line segment $\overline{\mathbf{xy}}$ lies in S and has $\mathrm{dist}(\overline{\mathbf{xy}}, \Gamma_S) \geq 2^{-n}$, then M rejects.
(b) If the line segment $\overline{\mathbf{xy}}$ does not lie in S,then M accepts.

This lemma implies the Π_1^P upper bound for the 1-path problem.

Corollary 3.2 *Assume that S has property (A) or property (B). Then, the 1-path problem of domain S is solvable by a Π_1^P oracle Turing machine (i.e., its complement is solvable by a polynomial-time nondeterministic oracle Turing machine).*

For $k \geq 2$, we can prove the Σ_2^P upper bound for the k-path problem. The idea of the proof is to nondeterministically guess $k-1$ breakpoints between the source point $\mathbf{x}$ and the target point $\mathbf{y}$, and then apply Lemma 3.1 to verify that each straight-line segment in the k-segment path formed by these points lies in the domain S. We omit the details of the proof.

Theorem 3.3 *Assume that $k \geq 2$ and S has property (A) or property (B). Then, there exist a polynomial-time nondeterministic oracle Turing machine M and a set $A \in NP$ such that, for any oracles $\phi_1, \phi_2, \phi_3, \phi_4$ representing two points $\mathbf{x}, \mathbf{y} \in S$, respectively, and for any input $n > 0$, the following conditions hold:*

(a) If there exists a k-segment path π from $\mathbf{x}$ to $\mathbf{y}$ such that π lies in domain S and $\mathrm{dist}(\pi, \Gamma_S) \geq 2^{-n}$, then $M^{A,\phi_1,\phi_2,\phi_3,\phi_4}(n)$ accepts.
(b) If there does not exist a k-segment path π from $\mathbf{x}$ to $\mathbf{y}$ that lies in domain S, then $M^{A,\phi_1,\phi_2,\phi_3,\phi_4}(n)$ rejects.

The oracle Turing machine M in Theorem 3.3 can be modified into a Turing transducer M_1 that outputs a k-segment path whenever M accepts.

Corollary 3.4 *Assume that $k \geq 2$ and S has property (A) or property (B). Then, there exist a polynomial-time nondeterministic oracle Turing transducer M_1 and a set $A \in NP$ such that, for any oracles $\phi_1, \phi_2, \phi_3, \phi_4$ representing two points $\mathbf{x}, \mathbf{y} \in S$, respectively, and for any input $n > 0$, the following conditions* *hold:*

(a) *If there exists a k-segment path π from $\mathbf{x}$ to $\mathbf{y}$ such that π lies in domain S and $\mathrm{dist}(\pi, \Gamma_S) \geq 2^{-n}$, then $M_1^{A,\phi_1,\phi_2,\phi_3,\phi_4}(n)$ accepts and outputs $k-1$ dyadic points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{k-1}$ in S such that the path consisting of $\overline{\mathbf{x}\mathbf{x}_1}$, $\overline{\mathbf{x}_1\mathbf{x}_2}, \ldots, \overline{\mathbf{x}_{k-2}\mathbf{x}_{k-1}}$, and $\overline{\mathbf{x}_{k-1}\mathbf{y}}$ is in S.*

(b) *If there does not exist a k-segment path π from $\mathbf{x}$ to $\mathbf{y}$ that lies in domain S, then $M_1^{A,\phi_1,\phi_2,\phi_3,\phi_4}(n)$ rejects.*

4 Lower Bounds

We first consider the 1-path problem. The proof of the following theorem is similar to, but simpler than, the proof of Theorem 4.3 (for the case $k \geq 2$). We omit the details.

Theorem 4.1 *For any set $A \in \Pi_1^P$, there exists a domain $S \subseteq [0,1]^2$ satisfying properties (A) and (B) such that A is polynomial-time computable relative to the solution of the 1-path problem of domain S.*

Together with Corollary 3.2, we obtain the following characterization of the complexity of the 1-path problem.

Corollary 4.2 *The following are equivalent:*

(a) $P = NP$.

(b) For any domain $S \subseteq [0,1]^2$ satisfying property (A) or (B), the corresponding 1-path problem is polynomial-time computable.

Next, we consider the k-path problems for $k \geq 2$. We show that, for each $k \geq 2$, there exists a domain S satisfying property (B) such that the k-path problem of domain S is Σ_2^P-hard.

Theorem 4.3 *Let $k \geq 2$. For any set $A \in \Sigma_2^P$, there exists a domain $S \subseteq [0,1]^2$ satisfying property (B) such that A is polynomial-time computable relative to the solution of the k-path problem of domain S.*

Proof. (Sketch). We first consider the case $k = 3$.

Recall that $A \in \Sigma_2^P$ means that there exist a set $B \in P$ and a polynomial function p such that, for any $w \in \{0,1\}^*$ of length n,

$$w \in A \iff (\exists u, |u| = p(n))\,(\forall v, |v| = p(n))\ \langle w, u, v\rangle \in B.$$

For any $w \in \{0,1\}^*$ of length $n > 0$, we will define a domain $T_w \subseteq [0,1]^2$. For any k, $0 \leq k \leq 2^{p(n)} - 1$, we let u_k be the kth string in $\{0,1\}^n$; that is,

u_k is the $p(n)$-bit binary representation of the integer k, with possible leading zeroes. We also let $a_k = k \cdot 2^{-p(n)}$ for any $0 \le k \le 2^{p(n)}$.

Now, for each $w \in \{0,1\}^*$ and each j, $0 \le j \le 2^{p(n)} - 1$, where $n = |w|$, we define a function $f_{w,j}$ that maps the unit interval $[0,1]$ to a curve in $[0,1]^2$. We divide the interval $[0,1]$ into $2^{p(n)}$ subintervals $[a_k, a_{k+1}]$, for $k = 0, 1, \ldots, 2^{p(n)} - 1$, and function $f_{w,j}$ maps each interval $[a_k, a_{k+1}]$ to either a line segment or a curve consisting of two line segments. Namely, for each k, $0 \le k \le 2^{p(n)} - 1$,

(i) $f_{w,j}$ maps $[a_k, a_{k+1}]$ to the line segment whose two endpoints are $\langle 1/4 + a_k/2, a_j\rangle$ and $\langle 1/4 + a_{k+1}/2, a_j\rangle$, if $\langle w, u_j, u_k\rangle \in B$; and

(ii) $f_{w,j}$ maps $[a_k, a_{k+1}]$ to a 2-segment path from $\langle 1/4 + a_k/2, a_j\rangle$ to $\langle 1/4 + a_{k+1}/2, a_j\rangle$, with the middle breakpoint $\langle 1/4 + a_k/2 + 2^{-p(n)-2}, a_j + 2^{-p(n)-1}\rangle$, if $\langle w, u_j, u_k\rangle \notin B$.

Let $\Gamma_{w,j}$ denote the image of $f_{w,j}$ on $[0,1]$. We show in Figure 1 the curve $\Gamma_{w,j}$, where $p(|w|) = 3$, $j = 0$, and $\langle w, u_j, u_k\rangle \in B$ for $k = 0, 3, 4, 5, 7$, and $\langle w, u_j u_k\rangle \notin B$ for $k = 1, 2, 6$. (In Figure 1, a_k^* denotes $1/4 + a_k$.)

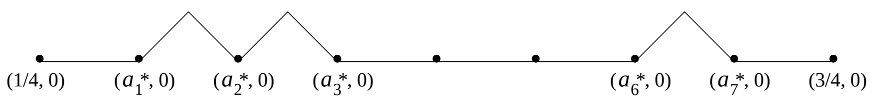

Fig. 1. A curve $\Gamma_{w,j}$.

Next, for each $w \in \{0,1\}^*$ and each j, $0 \le j \le 2^{p(n)}$, where $n = |w|$, we define a closed set $X_{w,j}$. First, let $R_{w,j,0}$ be the rectangle whose four corners are $\langle 1/8, a_{j-1} + 2^{-p(n)-2}\rangle$, $\langle 1/4, a_{j-1} + 2^{-p(n)-2}\rangle$, $\langle 1/4, a_j\rangle$ and $\langle 1/8, a_j\rangle$; and let $R_{w,j,1}$ be the rectangle whose four corners are $\langle 3/4, a_{j-1} + 2^{-p(n)-2}\rangle$, $\langle 7/8, a_{j-1} + 2^{-p(n)-2}\rangle$, $\langle 7/8, a_j\rangle$ and $\langle 3/4, a_j\rangle$. Now, set $X_{w,j}$ can be defined as follows:

(i) For $j = 0$, $X_{w,0}$ is the region enclosed between the line $\overline{\langle \frac{1}{4}, 0\rangle \langle \frac{3}{4}, 0\rangle}$ and the curve $\Gamma_{w,0}$.

(ii) For $1 \le j \le 2^{p(n)} - 1$, $X_{w,j}$ is the union of rectangles $R_{w,j,0}$, $R_{w,j,1}$ and the region enclosed between the line from $\langle 1/4, a_j - 2^{-p(n)-2}\rangle$ to $\langle 3/4, a_j - 2^{-p(n)-2}\rangle$ and $\Gamma_{w,j}$.

(iii) For $j = 2^{p(n)}$, $X_{w,j}$ is the union of three rectangles: $R_{w,j,0}$, $R_{w,j,1}$, and the rectangle whose four corners are $\langle 1/4, 1 - 2^{-p(n)-2}\rangle$, $\langle 3/4, 1 - 2^{-p(n)-2}\rangle$, $\langle 3/4, 1\rangle$ and $\langle 1/4, 1\rangle$.

We also define Y_w to be the union of two rectangles whose four corners are $\langle 7/8,\ 1 - 3 \cdot 2^{-p(n)-2}\rangle$, $\langle 15/16, 1 - 3 \cdot 2^{-p(n)-2}\rangle$, $\langle 15/16, 1\rangle$, $\langle 7/8, 1\rangle$ and, respectively, $\langle 15/16,\ 2^{-p(n)-2}\rangle$, $\langle 1, 2^{-p(n)-2}\rangle$, $\langle 1, 1\rangle$, $\langle 15/16, 1\rangle$. Fianlly, we let

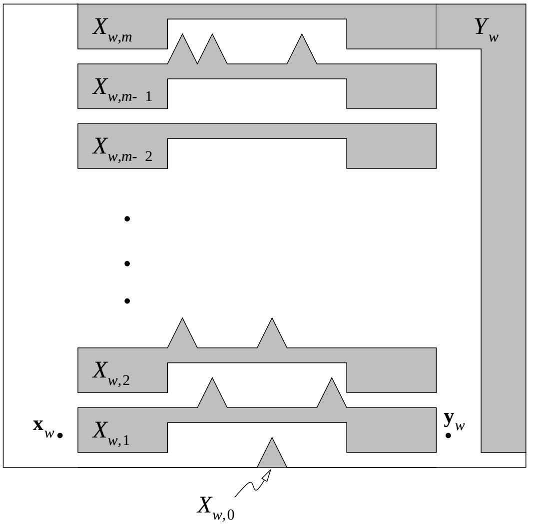

Fig. 2. The domain T_w, where $m = 2^{p(n)}$.

$T_w = [0,1]^2 - \bigcup_{j=0}^{2^{p(n)}} X_{w,j} - Y_w$, and $\mathbf{x}_w = \langle 1/8 - 2^{-p(n)-2}, 2^{-p(n)-1}\rangle$, $\mathbf{y}_w = \langle 7/8 + 2^{-p(n)-2}, 2^{-p(n)-1}\rangle$ (see Figure 2).

We observe that the domain T_w and points $\mathbf{x}_w$, $\mathbf{y}_w$ satisfy the following properties:

(1) If there exists an integer j, $0 \le j \le 2^{p(n)} - 1$ such that for all k, $0 \le k \le 2^{p(n)} - 1$, $\langle w, u_j, u_k\rangle \in B$, then there is a 3-segment path in T_w from $\mathbf{x}_w$ to $\mathbf{y}_w$, whose middle two breakpoints are $\langle 1/8 - 2^{-p(n)-2}, a_j + 2^{-p(n)-3}\rangle$ and $\langle 7/8 + 2^{-p(n)-2}, a_j + 2^{-p(n)-3}\rangle$.

(2) Otherwise, all $2^{p(n)}$ possible pathways are blocked by some bumps of the functions $f_{w,j}$, and so there is no 3-segment path in T_w from $\mathbf{x}_w$ to $\mathbf{y}_w$.

Now, we can combine domains T_w into a single domain S. For each $n > 0$, let $c_n = 1 - 2^{-(n-1)}$. We define, for each $w \in \{0,1\}^*$ of length n, domain S_w to be the image of domain T_w under the linear transformation

$$g_w(\langle x_1, x_2\rangle) = \langle c_n + x_1 \cdot 2^{-n}, (k_w + x_2) \cdot 2^{-n}\rangle,$$

where k_w is the integer whose n-bit binary representation (with possible leading zeroes) is exactly w. Let $S = \bigcup_{w\in\{0,1\}^+} S_w$, (see Figure 3).

It is not hard to verify that S satisfies property (B). That is, for any given point $\mathbf{z} \in [0,1]^2$ and integer $n > 0$, if $\operatorname{dist}(\mathbf{z}, \Gamma_S) \ge 2^{-n}$, then we can determine whether $\mathbf{z} \in S$ and find a dyadic rational d such that $|d - \operatorname{dist}(\mathbf{z}, \Gamma_S)| \le 2^{-n}$.

In addition, from the above observations (1) and (2) about T_w and $\mathbf{x}_w$, $\mathbf{y}_w$,

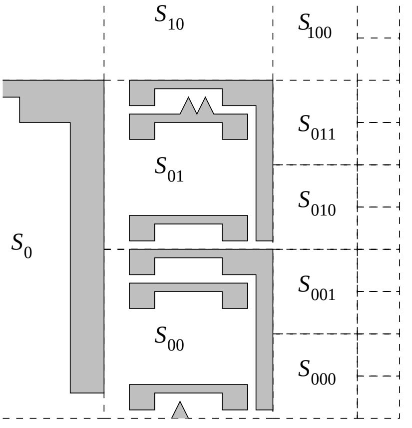

Fig. 3. The domain S.

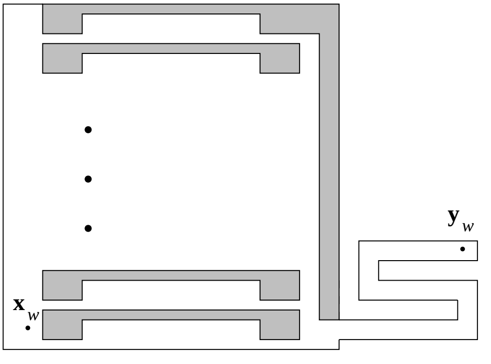

Fig. 4. The domains T_w in the case of $k = 8$.

we see that $w \in A$ if and only if there is a 3-segment path between $g_w(\mathbf{x}_w)$ and $g_w(\mathbf{y}_w)$. So, the question of whether $w \in A$ can be solved in polynomial time by asking whether there is a 3-segment path in S from $g_w(\mathbf{x}_w)$ to $g_w(\mathbf{y}_w)$). This completes the proof of the case $k = 3$.

For $k > 3$ and $k = 2$, we can modify the above construction to get the required domain S. We omit the details, and only show the domain T_w for the case of $k = 8$ in Figure 4, and the domain T_w for the case $k = 2$ in Figure 5. □

Corollary 4.4 *Let $k \geq 2$. The following are equivalent:*

(a) $\Sigma_2^P = \Sigma_1^P$.

(b) For any domain $S \subseteq [0,1]^2$ satisfying property (B), the corresponding k-path problem is nondeterministic polynomial-time computable.

We remark that it is known in discrete complexity theory that $P = NP$

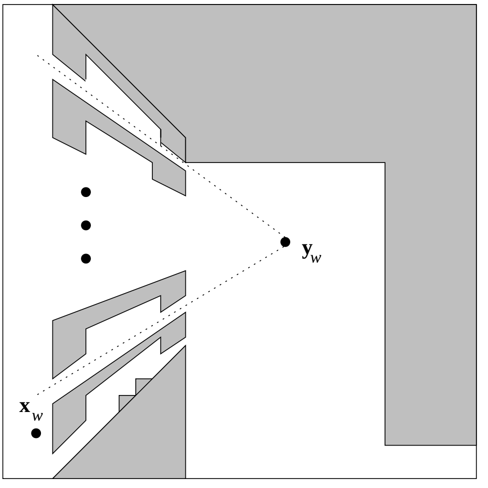

Fig. 5. The domains T_w in the case of $k = 2$.

implies $\Sigma_2^P = \Sigma_1^P = P$ (see, e.g., Du and Ko [4]). Therefore, Corollary 4.2 also holds for k-path problems with $k \geq 2$.

Corollary 4.5 *Let $k \geq 2$. The following are equivalent:*

(a) $P = NP$.

(b) For any domain $S \subseteq [0, 1]^2$ satisfying property (B), the corresponding k-path problem is polynomial-time computable.

5 Final Remarks

We observe that the domain S constructed in Theorem 4.3, though satisfying property (B), is not a simply connected domain. In fact, it contains an infinite number of holes. Whether this construction can be strengthened to make S a simply connected domain, or a multiply connected domain (i.e., a domain with a finite number of holes), is an interesting open question. We do know that the result of Theorem 4.1 also applies to the k-path problem for $k > 1$. So, we have the following weaker lower bound for simply connected domains S satisfying either property (A) or property (B).

Corollary 5.1 *Let $k \geq 2$. For any set $A \in \Pi_1^P$, there exists a simply connected domain $S \subseteq [0, 1]^2$ satisfying properties (A) and (B) such that A is polynomial-time computable relative to the solution of the k-path problem of domain S.*

In addition, we observe that the holes in domain S constructed in Theorem 4.3 can be eliminated if we consider *three-dimensional domains*.

Corollary 5.2 *Let $k \geq 2$. For any set $A \in \Sigma_2^P$, there exists a simply connected three-dimensional domain $S \subseteq [0,1]^3$ satisfying properties (A) and (B), such that A is polynomial-time computable relative to the solution of the k-path problem of domain S.*

Sketch of Proof. We describe the idea for the case $k = 3$. Consider the construction of domain T_w in Theorem 4.3. We can think of T_w as the unit square $[0,1]^2$ with a number of holes $X_{w,j}$, $0 \leq j \leq 2^{p(n)}$, and Y_w. Now, we start with the unit cube $[0,1]^3$, and for each j, $0 \leq j \leq 2^{p(n)}$, we dig a hole (from top) of the shape $X_{w,j}$ of depth $1/2$, and a hole of the shape Y_w of the same depth. Call the remaining region U_w (which contains the bottom half of the cube and so is simply connected). In addition, attach two bars $B_{w,1} = [0,1/8] \times [-1,0] \times [7/8,1]$ and $B_{w,2} = [7/8,1] \times [-1,0] \times [7/8,1]$ to U_w. Let T'_w be the interior of $U_w \cup B_{w,1} \cup B_{w,2}$, and let $\mathbf{x}'_w = \langle 1/16, -15/16, 15/16 \rangle$ and $\mathbf{y}_w = \langle 15/16, -15/16, 15/16 \rangle$ (so that $\mathbf{x}_w$ and $\mathbf{y}_w$ are close to the end of the two bars $B_{w,1}$ and $B_{w,2}$, respectively). We note that any path from $\mathbf{x}'_w$ to $\mathbf{y}'_w$ must contain first a line segment from $B_{w,1}$ to U_w, and at last a line segment from U_w to $B_{w,2}$, and so the only possible 3-segment path from $\mathbf{x}'_w$ to $\mathbf{y}'_w$ must stay in the top half of the domain T'_w, and so satisfies the conditions (1) and (2) of the proof of Theorem 4.3.

With some care, we can combine domains T'_w into a single domain S' that is simply connected, and satisfies property (B). Actually, S' also satisfies property (A), since each hole $X_{w,j}$ in the original domain T_w can be computed in polynomial time in $|w|$. □

References

[1] Chou, A. W. and Ko, K., Computational complexity of two-dimensional regions, *SIAM J. Comput.* **24** (1995), 923–947.

[2] Chou, A. W. and Ko, K., On the complexity of finding paths in a two-dimensional domain I: shortest paths, *Math. Logic Quarterly*, to appear; also in *Proc. International Conference on Computability and Complexity in Analysis*, 2003, Informatik Berichte 302-8/2003, FernUniversität in Hagen, Hagen, Germany.

[3] Chou, A. W. and Ko, K., A note on the complexity of distance functions of two-dimensional domains, *preprint*.

[4] Du, D.-Z. and Ko, K., *Theory of Computational Complexity*, John Wiley & Sons, New York, 2000.

[5] Guibas, L., Hershberger, J., Leven, D., Sharir, M. and Tarjan, R., Linear time algorithms for visibility and shortest path problems inside simple polygons, *Proceedings of the 2nd ACM Symposium on Computational Geometry*, ACM, New York, 1986, pp. 1–13.

[6] Guibas, L. and Hershberger, J., Optimal shortest path queries in a simple polygon, *Proceedings of the 3rd ACM Symposium on Computational Geometry*, ACM, New York, 1987, pp. 50–63.

[7] Henrici, P., *Applied and Computational Complex Analysis*, Vol. 1–3. John Wiley & Sons, New York, 1974, 1977, 1986.

[8] Hopcroft, J. E., Schwartz, J. T. and Sharir, M., *Planning, Geometry, and Complexity of Robot Motion*, Ablex, Norwood, NJ, 1987.

[9] Ko, K., *Complexity Theory of Real Functions*, Birkhäuser, Boston, 1991.

[10] Ko, K., Polynomial-time computability in analysis, in *Handbook of Recursive Mathematics, Vol. 2: Recursive Algebra, Analysis and Combinatorics*, Yu. L. Ershov et al., eds., Studies in Logic and the Foundations of Mathematics, Vol. 139, Elsevier, Amsterdam, 1998, pp. 1271–1317.

[11] Ko, K. and Friedman, H., Computational complexity of real functions, *Theoret. Comput. Sci.* **20** (1982), 323–352.

[12] Mitchell, J. and Sharir, M., New results on shortest paths in three dimensions, *Proc. 20th ACM Symposium on Computational Geometry*, ACM, NY, 2004 (to appear).

[13] Pour-El, M. and Richards, I., *Computability in Analysis and Physics*, Springer-Verlag, Berlin , 1989.

[14] Reif, J.H. and Storer, J.A., Minimizing turns for discrete movement in the interior of a polygon, *IEEE J. Robot. Autom.*, **3** (1987), 182–193.

[15] Storer, J.A., On minimal node-cost planar embeddings, *Networks* **14** (1984), 181–212.

[16] Suri, S., A linear time algorithm for minimum link path in a simple polygon, *Comput. Vision, Graphics, Image Proc.* **35** (1986), 99-110.

[17] Tamassia, R., On embedding a graph in the grid with the minimum number of bends, *SIAM J. Comput.* **16** (1987) 421–444.

[18] Weihrauch, K., *Computable Analysis*, Springer-Verlag, Heidelberg, 2000.