On the Complexity of Finding Circumscribed Rectangles and Squares for a Two-Dimensional Domain *

Fuxiang Yu[†] Arthur Chou[‡] Ker-I Ko[§]

Abstract

We investigate the computational complexity of finding the minimum-area circumscribed rectangle of a given two-dimensional domain. We study this problem in the polynomial-time complexity theory of real functions based on the oracle Turing machine model. We show that a bounded domain S with a polynomial-time computable Jordan curve Γ as the boundary may not have a computable minimum-area circumscribed rectangle. We also show that the problem of finding the minimum area of a circumscribed rectangle of a polynomial-time computable Jordan curve Γ is equivalent to a discrete Σ_2^P-complete problem. The related problem of finding the circumscribed squares of a Jordan curve Γ is also studied. We show that for any polynomial-time computable Jordan curve Γ, there must exist at least one computable circumscribed square (not necessarily of the minimum area), but this square may have arbitrarily high complexity.

1 Introduction

Let $S \subseteq \mathbb{R}^2$ be a bounded, connected domain in the two-dimensional plane. How do we find a circumscribed rectangle of S with the minimum area? This is a basic problem in computational geometry with applications in computer graphics and robotics (see, e.g., Schneider and Eberly [17], Freeman and Shapiro [8] and Toussaint [19]). In this paper, we investigate this problem from the viewpoint of computational complexity. More precisely, we assume that the boundary of set S is a polynomial-time computable Jordan curve Γ (i.e., Γ has a polynomial-time representation $f : [0, 1] \to \mathbb{R}^2$), and ask what the complexity is of the minimum-area circumscribed rectangle of S.[1] We study this problem in the context of complexity theory of real functions of Ko and Friedman [11], which uses oracle Turing machines as the basic computational model, and defines the complexity in terms of the precision of the output values of the functions under consideration.

---

[*] This material is based upon work supported by National Science Foundation under grant No. 0430124.

[†] Department of Computer Science, State University of New York at Stony Brook, Stony Brook, NY 11794 (fuxiang@cs.sunysb.edu).

[‡] Department of Mathematics and Computer Science, Clark University, Worcester, Massachusetts 01610 (achou@clarku.edu).

[§] Department of Computer Science, State University of New York at Stony Brook, Stony Brook, NY 11794 (keriko@cs.sunysb.edu).

[1] In the rest of the paper, we write "the minimum circumscribed rectangle" to mean "the minimum-area circumscribed rectangle".

Let Γ be a Jordan curve on the two-dimensional plane $\mathbb{R}^2$. For any line L_α that forms an angle α with the x-axis, there is a unique rectangle R_α that circumscribes the curve Γ and has two of its four sides parallel to L_α. Thus, the minimum circumscribed rectangle problem may be viewed as the problem of finding an angle α that minimizes the area of R_α. This problem is similar to the general minimization problem studied in Ko [10], which asks for the minimum value of a polynomial-time computable real function. The main difference here is that the underlying function mapping α to rectangle R_α is not necessarily polynomial-time computable. Our main results, following this direction, can be summarized as follows:

(1) For a fixed polynomial-time computable angle α, we can always find the circumscribed rectangle R_α of a polynomial-time computable Jordan curve Γ in polynomial time if and only if $P = NP$.

(2) There exists a polynomial-time computable Jordan curve Γ such that it has an uncountable number of minimum circumscribed rectangles, but none of them is computable (cf. the result on roots in Specker [18]).

(3) If a Jordan curve Γ is polynomial-time computable, then the area V of the minimum circumscribed rectangle R_α of Γ is a right Σ_2^P-real number (See Section 2 for the definition).

(4) There exists a polynomial-time computable Jordan curve Γ such that the problem of finding the area $V_{a,b}$ of the minimum circumscribed rectangle R_α of Γ, with the restriction of $a \le \alpha \le b$, is Σ_2^P-hard.

In addition to these results, we also study the problem of finding the minimum circumscribed square of a Jordan curve Γ. It is interesting to point out that this problem is not quite the same as the problem of finding the minimum circumscribed rectangle of Γ. In fact, a minimum square that encloses a Jordan curve Γ is not necessarily a circumscribed square of Γ. In addition to the extension of result (2) above for the minimum circumscribed squares, we also show that for any polynomial-time computable Jordan curve Γ, there must exist at least one computable circumscribed square (not necessarily of the minimum area), but this square may have arbitrarily high complexity.

Our basic computational model for real-valued functions and two-dimensional regions is the oracle Turing machine. For the general theory of computable analysis based on the Turing machine model, see, for instance, Pour-El and Richards [14] and Weihrauch [20]. For the theory of computational complexity of real functions based on this computational model, see Ko [10]. The extension of this theory to include the computational complexity of two-dimensional regions has been presented in Chou and Ko [3]. Computational complexity of problems related to two-dimensional regions has been studied recently in several directions. Rettinger and Weihrauch [16], Braverman [1], Rettinger [15] and Braverman et al. [2] studied the the computational complexity of Julia sets. Chou and Ko [4, 5] studied the problem of finding paths in a two-dimensional domain. Ko and Yu [12] studied the problem of computing single-valued analytic branches of logarithm and square-root functions on a two-dimensional domain. All these works used Turing machines and oracle Turing machines as the basic model.

2 Definitions and Notation

The fundamental *discrete* complexity classes we are interested in are the class P of sets accepted by deterministic polynomial-time Turing machines, and the class NP of sets accepted by nondeterministic polynomial-time Turing machines. In this paper, we are also interested in the complexity classes Σ_2^P and Π_2^P in the second level of the polynomial-time hierarchy. To be more precise, the class Σ_2^P consists of all sets that can be solved by a polynomial-time nondeterministic oracle Turing machine using a set $A \in NP$ as an oracle, and the class Π_2^P consists of all sets whose complement is in Σ_2^P. Equivalently, a set $A \subseteq \{0,1\}^*$ is in Σ_2^P if there exist a polynomial function p and a polynomial-time computable predicate Q such that, for every $w \in \{0,1\}^*$ of length $n \geq 0$, the following holds:[2]

$$w \in A \iff (\exists u, \ell(u) \leq p(n))\, (\forall v, \ell(v) \leq p(n))\, Q(w,u,v).$$

It is known that if $P = NP$ then $\Sigma_2^P = \Pi_2^P$, but the converse is not known. See, for instance, Du and Ko [7], for more properties about the complexity classes Σ_2^P and Π_2^P.

The basic computational objects in the continuous computation in the Turing machine model are dyadic rationals $\mathbb{D} = \{m/2^n : m \in \mathbb{Z}, n \in \mathbb{N}\}$. Each dyadic rational d has infinitely many binary representations, with arbitrarily many trailing zeros. For each $n \in \mathbb{N}$, we let $\mathbb{D}_n$ denote the class of dyadic rationals which have a binary representation of at most n bits to the right of the binary point; that is, $\mathbb{D}_n = \{m/2^n : m \in \mathbb{Z}\}$.

A real number has a number of representations. The most basic one is the *Cauchy function representation.* We say a function $\phi : \mathbb{N} \to \mathbb{D}$ *binary converges* to a real number x, or is *a Cauchy function representation of* x, if (i) for all $n \geq 0$, $\phi(n) \in \mathbb{D}_n$, and (ii) for all $n \geq 0$, $|x - \phi(n)| \leq 2^{-n}$. For any $x \in \mathbb{R}$, there is a unique function $\phi_x : \mathbb{N} \to \mathbb{D}$ that binary converges to x and satisfies the condition $x - 2^{-n} < \phi_x(n) \leq x$ for all $n \geq 0$. We call this function ϕ_x the *standard Cauchy function* of x.

For each Cauchy function representation $\phi : \mathbb{N} \to \mathbb{D}$ of a real number x, there is an associated (*general*) *left cut representation*, namely, the set $L_\phi = \{d \in \mathbb{D}_n : d \leq \phi(n), n \geq 1\}$. Note that the set L_ϕ also uniquely determines the Cauchy function ϕ since $\phi(n) = \max\{\mathbb{D}_n \cap L_\phi\}$ for all $n \geq 1$. In other words, for a real number x, there is a 1-1 correspondence between the set of its Cauchy function representations and the set of its left cuts. The (general) left cut L_{ϕ_x} of x associated with the standard Cauchy function ϕ_x of x is called the *standard left cut* of x. Note that $L_{\phi_x} = \{d \in \mathbb{D} : d \leq x\}$.

The computability and complexity of a real number can be defined according to the computability and complexity of its Cauchy function and left cut representations. It is natural to use the Cauchy function representation of a real number x since it gives the approximations directly. On the other hand, the left cut representation of a real number x is also useful when we consider complexity classes $\mathcal{C}$ of sets, such as NP and Σ_2^P, instead of complexity classes of functions.

Definition 2.1 (*a*) *A real number x is said to be* computable *if it has a computable Cauchy function representation or, equivalently, if it has a computable left cut.*

---

[2] We write $\ell(u)$ to denote the length of a string u. The notation $|x|$ is reserved for the absolute value of a real or complex number x.

(*b*) *Let* $\mathcal{C}$ *be a complexity class of sets. A real number* x *is said to be a* left $\mathcal{C}$-real number *if it has a left cut in* $\mathcal{C}$, *and* x *is a* right $\mathcal{C}$-real number *if it has a* right cut (*the complement of a left cut*) *in* $\mathcal{C}$.

The prefixes *left* and *right* in the above definition are necessary since, for many complexity classes $\mathcal{C}$ (e.g., $\mathcal{C} = NP$ or Σ_2^P), it is not known whether the complement *co*-$\mathcal{C}$ of $\mathcal{C}$ is identical to $\mathcal{C}$ or not. As $P =$ *co*-P, we say a number x is a P-real number, or x is *polynomial-time computable*, if x is a left or right P-real number. We remark that a real number x is polynomial-time computable iff it has a polynomial-time computable Cauchy function representation since we can compute, by a simple binary search, the Cauchy function ϕ from its associated left cut L_ϕ in polynomial time.

To compute a real-valued function $f : [0,1]\to\mathbb{R}$, we use oracle Turing machines as the computational model.

Definition 2.2 (*a*) *A real function* $f : [0,1]\to\mathbb{R}$ *is said to be* computable *if there exists an oracle* (*Turing*) *machine* M *such that for any oracle* ϕ *that binary converges to a real number* $x \in [0,1]$, *the set* $\{d \in \mathbb{D} : M^\phi(d)$ *accepts*$\}$ *is a general left cut of* $f(x)$.

(*b*) *If the oracle Turing machine* M *in* (*a*) *is a polynomial-time deterministic oracle machine* (*i.e., if there exists a polynomial function* p *such that for any* $n \geq 1$ *and any* $d \in \mathbb{D}_n$, $M^\phi(d)$ *halts in time* $p(n)$), *then* f *is said to be* polynomial-time computable.

(*c*) *If the oracle machine* M *in* (*a*) *is a polynomial-time nondeterministic oracle machine, then* f *is said to be* computable in *NP*-time, *or simply an* *NP*-real function.

We remark again that computable and polynomial-time computable functions can also be defined, equivalently, using the Cauchy function representation. For instance, a function $f : [0,1]\to\mathbb{R}$ is polynomial-time computable by Definition 2.2, if and only if there exist a polynomial function p and an oracle Turing machine M such that, for any oracle ϕ that binary converges to a real number $x \in [0,1]$ and for any input $n > 0$ (which serves as a precision parameter), $M^\phi(n)$ halts in time $p(n)$ and outputs a dyadic rational e such that $|e - f(x)| \leq 2^{-n}$.

The complexity of the maximum value of a real function has been studied in Ko [10]. The following results are related to our study here.

Proposition 2.3 (*a*) *A real number* x *is a left NP-real number if and only if there is a polynomial-time computable real function* $f : [0,1]\to\mathbb{R}$ *such that* $x = \max_{0\leq t\leq 1} f(t)$.

(*b*) *A real number* x *is a left* Π_2^P*-real number if and only if there is an NP-real function* $f : [0,1]\to\mathbb{R}$ *such that* $x = \min_{0\leq t\leq 1} f(t)$.

Definition 2.2 can be extended naturally to functions $f : \mathbb{R}\to\mathbb{R}^2$ and functions $f : \mathbb{R}^2\to\mathbb{R}^2$. A Jordan curve (simple, closed curve) Γ in $\mathbb{R}^2$ is *polynomial-time computable* if there exists a polynomial-time computable function $f : [0,1]\to\mathbb{R}^2$ such that the range of f is Γ, f is one-to-one on $[0,1)$ and $f(0) = f(1)$.

3 Minimum Circumscribed Rectangles

We first give a formal definition of circumscribed polygons of a Jordan curve.

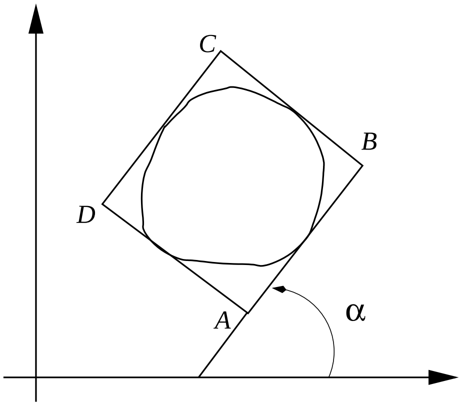

Figure 1: A circumscribed rectangle of a Jordan curve

Definition 3.1 *We say a polygon T circumscribes a Jordan curve Γ if* (1) *every point of Γ lies in the interior of T or on T, and* (2) *Γ intersects every side of T.*

As pointed out in Section 1, for a Jordan curve Γ and a fixed angle $\alpha \in [0, \pi/2)$, there is a unique rectangle R_α that circumscribes the curve Γ and has two of its four sides forming angle α with the x-axis (see Figure 1); we call R_α the circumscribed rectangle of Γ at angle α. We first consider the computational complexity of the rectangle R_α, when α is a fixed polynomial-time computable real number. We say a rectangle R is *polynomial-time computable* if its four corners are polynomial-time computable complex numbers.

Without loss of generality, we may assume that the circumscribed rectangle under consideration is rectangle R_0 at angle $\alpha = 0$, which has two horizontal and two vertical sides. We note that the top horizontal side of this rectangle R_0 is $y = u$, where u is the maximum y-value of the curve Γ, and the bottom horizontal line of R_0 is $y = b$, where b is the minimum y-value of the curve Γ. Since the curve Γ is polynomial-time computable, the values u and b are just the maximum and minimum values of a polynomial-time computable function, respectively.

From the above observation, the characterization of maximum and minimum values of Proposition 2.3 can be applied to the complexity of the rectangle R_0.

Theorem 3.2 (a) *Let Γ be a polynomial-time computable Jordan curve, and R_0 the circumscribed rectangle of Γ at angle $\alpha = 0$. Let the four sides of the rectangle R_0 be $y = u$, $y = b$, $x = \ell$ and $x = r$, with $u > b$ and $r > \ell$. Then, u and r are left NP-real numbers, and b and ℓ are right NP-real numbers.*

(b) *For any left NP-real number u, there is a polynomial-time computable Jordan curve Γ such that the top horizontal side of its circumscribed rectangle R_0 at angle $\alpha = 0$ is $y = u$.*

Proof. (a) Let f be a polynomial-time computable function that represents a Jordan curve Γ, and $f_x(t)$, $f_y(t)$ be two functions mapping $[0, 1]$ to $\mathbb{R}$ such that $f(t) = \langle f_x(t), f_y(t)\rangle$ for $t \in [0, 1]$. Then, both f_x and f_y are polynomial-time computable. It is clear that the top side of R_0 is $y = \max_{t\in[0,1]} f_y(t)$, and the bottom side of R_0 is $y = \min_{t\in[0,1]} f_y(t)$. Similarly, the right side of R_0 is $x = \max_{t\in[0,1]} f_x(t)$, and the left side of R_0 is $x = \min_{t\in[0,1]} f_x(t)$.

Note that, for any function $g : [0,1] \to \mathbb{R}$, $\min_{t \in [0,1]} g(t) = -\max_{t \in [0,1]}(-g(t))$. Also note that, for any left cut L of a real number x, the set $\{-d \mid d \in L\}$ is a right cut of $-x$. Thus, part (a) of the theorem follows from Proposition 2.3.

(b) Assume that $u > 0$. Let $g : [0,1] \to \mathbb{R}$ be a polynomial-time computable function with $\max_{t \in [0,1]} g(t) = u$, as given by Proposition 2.3. Without loss of generality, we may assume that (a) $g(0) = g(1) = 0$, and (b) $g(t) > 0$ for all $t \in (0,1)$.

Define a function $f : [0,1] \to \mathbb{R}^2$ as follows: f on $[0,1/2]$ is the line segment from the point $\langle 1,0 \rangle$ to the point $\langle 0,0 \rangle$; and $f(t) = \langle 2t-1, g(2t-1) \rangle$ for $t \in (1/2,1]$. Then, it is clear that f defines a polynomial-time computable Jordan curve Γ and the circumscribed rectangle R_0 of Γ at angle $\alpha = 0$ is formed by the following four lines: $y = u$, $y = 0$, $x = 1$, and $x = 0$. □

A set $A \subseteq \{0\}^*$ of strings formed by a singleton alphabet is called a *tally set*. Let P_1 and NP_1 denote the class of tally sets in P and NP, respectively. It is known that if $P_1 \neq NP_1$ then there exists a left *NP*-real number which is not polynomial-time computable (see Ko [10]).

Corollary 3.3 *In the following,* (*a*)⇒(*b*)⇒(*c*)*:*

(*a*) $P = NP$.

(*b*) *For every polynomial-time computable Jordan curve, its circumscribed rectangle* R_0 *at angle* $\alpha = 0$ *is polynomial-time computable.*

(*c*) $P_1 = NP_1$.

Corollary 3.4 (*a*) *For any polynomial-time computable Jordan curve* Γ *and any polynomial-time computable real number* $\alpha \in [0, \pi/2)$*, the height, width, and area of the circumscribed rectangle* R_α *of* Γ *at angle* α *are left NP-real numbers.*

(*b*) *If* $P_1 \neq NP_1$*, then there exists a polynomial-time computable Jordan curve* Γ *such that the area of its circumscribed rectangle* R_0 *at angle* $\alpha = 0$ *is not polynomial-time computable.*

Proof. We note that the sum of two left *NP*-real numbers is still a left *NP*-real number, and the product of two positive left *NP*-real numbers is still a left *NP*-real number. □

We are interested in the minimum circumscribed rectangle of a Jordan curve Γ. Since the curve Γ has a unique circumscribed rectangle R_α at each angle $\alpha \in [0, \pi/2)$, this is essentially the problem of finding the angle α that minimizes the area of R_α. Ko [10] pointed out that the computability of the maximum points of a computable function $g : [0,1] \to \mathbb{R}$ is very similar to the computability of its roots. In particular, the result of Specker [18] about roots also holds for the maximum points; that is, there exists a computable function $g : [0,1] \to \mathbb{R}$ such that it has an uncountable number of maximum points but none of them is computable. Furthermore, this result holds even if g is required to be polynomial-time computable.

Theorem 3.5 *There exists a polynomial-time computable Jordan curve* Γ *such that* Γ *has an uncountable number of minimum circumscribed rectangles but none of them is computable.*

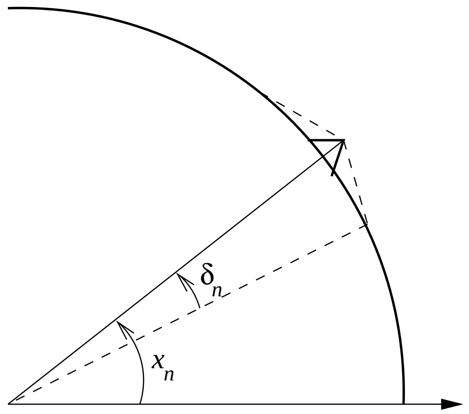

Figure 2: The construction of Theorem 3.5

Sketch of Proof. Since the proof follows the idea of Specker's theorem, we only present a sketch of the construction. We first let Γ_0 be the unit circle. That is, Γ_0 is the image of $f_0(\alpha) = \langle \cos\alpha, \sin\alpha\rangle$ on $[0, 2\pi]$. Note that all circumscribed rectangles of Γ_0 have the same area 4.

Now, let $\{x_n\}_{n=0}^{\infty}$ be a recursive enumeration of all computable real numbers in $[0, \pi/2]$. For each $n \geq 0$, we add to the circle Γ_0 a small Λ-shaped bump at angle x_n, of the same width and height $h_n = 2^{-k(n+t(n))}$, where $k \geq 2$ and $t(n)$ is the time required to enumerate the n-th number x_n (see Figure 2). Let Γ be the resulting curve. Since the height h_n of the bump is smaller than $2^{-t(n)}$, the curve Γ remains polynomial-time computable.

Let $\delta_n = \arccos(1/(1 + h_n))$. Then, the area of a circumscribed rectangle R_α of Γ at angle $\alpha \in (x_n - \delta_n, x_n + \delta_n)$ is greater than 4. By choosing a sufficiently large k, we can ensure that the sum $\sum_{n\geq 0} 2\delta_n$ is less than $\pi/2$. Therefore, the set

$$T = [0, \pi/2] - \bigcup_{n\geq 0} (x_n - \delta_n, x_n + \delta_n)$$

is nonempty and has a positive measure. In addition, for each $\beta \in T$, the circumscribed rectangle R_β of Γ at angle β remains the same with area 4. Finally, for each $\beta \in T$, one of the vertices of the circumscribed rectangle R_β at angle β is $\langle \sqrt{2}\cdot\cos(\beta - \pi/4), \sqrt{2}\cdot\sin(\beta - \pi/4)\rangle$, and so R_β is not a computable rectangle. □

Now we consider the problem of finding the minimum area of a circumscribed rectangle of a polynomial-time computable Jordan curve. First note that Theorem 3.2(a) can be extended so that the area $v(\alpha)$ of the circumscribed rectangle R_α of a polynomial-time computable Jordan curve at an angle α is actually an *NP-real function.* Thus, the problem of finding the minimum area of a circumscribed rectangle of Γ is just to find the minimum value of an *NP*-real function. Proposition 2.3(b) suggests that the complexity of the minimum area $v(\alpha)$ is a right Σ_2^P-real number (i.e., a left Π_2^P-real number).

Theorem 3.6 *Let Γ be a polynomial-time computable Jordan curve. Then, for any dyadic rational numbers $0 \leq a < b \leq \pi/2$, the minimum area of a circumscribed rectangle R_α of Γ at angle $\alpha \in [a, b]$ is a right Σ_2^P real number.*

Proof. Assume that Γ is computed by a real function $f : [0,1]\to\mathbb{R}^2$ in time $p(n)$ for some polynomial p. Let R_α be a minimum-area circumscribed rectangle of Γ at angle α, and $v(\alpha)$ be its area. Without loss of generality, assume that $R_\alpha \subseteq [0,1]^2$. Suppose d is a dyadic rational in $\mathbb{D}_n$ which is greater than or equal to $v(\alpha)$. Let e be a dyadic rational in $\mathbb{D}_{n+4}$ such that $|\alpha - e| \leq 2^{-(n+4)}$. Then, the area of the circumscribed rectangle R_e of Γ at angle e is less than $v(\alpha) + 4 \cdot 2^{-(n+4)}$. Let S_e be a rectangle that is parallel to and encloses the rectangle R_e, with the following properties:

(a) The upper right corner of S_e is a dyadic rational point $\mathbf{z}$ in $(\mathbb{D}_{n+5})^2$,

(b) The height d_h and width d_w of S_e are two dyadic rationals in $\mathbb{D}_{n+5}$, and

(c) The distance between the corresponding sides of R_e and S_e is between $2^{-(n+5)}$ and $2^{-(n+4)}$.

Then, the area S_e is $d_h \cdot d_w \leq area(R_e) + 4 \cdot 2^{-(n+4)} < d + 2^{-(n+1)}$.

We now design a Σ_2^P machine (a polynomial-time nondeterministic oracle machine using a discrete oracle set $A \in NP$ as the oracle) M to accept a right cut of $v(\alpha)$ based on the properties of rectangle S_e:

> Input: $d \in \mathbb{D}_n$.
>
> The machine M first nondeterministically guesses a dyadic rational point $\mathbf{z}$ in $(\mathbb{D}_{n+5})^2$, a dyadic rational $e \in \mathbb{D}_{n+4}$ and two dyadic rationals d_h, d_w in $\mathbb{D}_{n+5}$. Then, M forms the rectangle S_e from $\mathbf{z}, e, d_h, d_w$ as discussed above, and verifies that the curve Γ is inside S_e. More precisely, the verification can be done as follows: For every $t \in [0,1]\cap\mathbb{D}_{p(n+5)}$, get a dyadic point y_t that is within distance $2^{-(n+5)}$ of the point $f(t)$, and verify that y_t lies inside the rectangle S_e (note that $f(t) \in R_e \Rightarrow y_t \in S_e$ if property (c) above holds). In other words, M uses the set $A = \{\langle \mathbf{z}, e, , d_h, d_w\rangle : (\exists t \in \mathbb{D}_{p(n+5)})\, y_t \notin S_e\}$ as an oracle, and accepts d if $\langle \mathbf{z}, e, d_h, d_w\rangle \notin A$ and $d_h d_w < d + 2^{-(n+1)}$.

It is clear that the rectangle S_e is uniquely defined by $\mathbf{z}, e, d_h$ and d_w, and it can be determined in polynomial time whether a dyadic point $\mathbf{x}$ is in S_e or not. Thus, set A is in NP, and M is a Σ_2^P machine.

The analysis given above shows that this Σ_2^P machine accepts d if $d \geq v(\alpha)$. In addition, if M accepts d, then there must be a rectangle S_e of area $d_h d_w < d + 2^{-(n+1)}$ such that all points in Γ are either inside S_e or within distance $2^{-(n+5)}$ of the boundary of S_e. So, the minimum area $v(\alpha)$ is less than $d + 2^{-(n+1)} + 4 \cdot 2^{-(n+5)} < d + 2^{-n}$. This means that if $d \leq v(\alpha) - 2^{-n}$, then M rejects it. It follows that M accepts a right cut of $v(\alpha)$. □

Next we consider the converse of the above theorem. Let Γ be a polynomial-time computable Jordan curve. We will show that the general question of finding the minimum area of a circumscribed rectangle of Γ at an angle between a given range $[a,b]$ is Σ_2^P-hard.

Recall that a set $A \subseteq \{0,1\}^*$ is in Σ_2^P if and only if there exist a polynomial function p and a polynomial-time computable predicate Q such that, for all $w \in \{0,1\}^*$,

$$w \in A \Leftrightarrow (\exists u, \ell(u) = p(\ell(w)))(\forall v, \ell(v) = p(\ell(w)))Q(w,u,v). \tag{3.1}$$

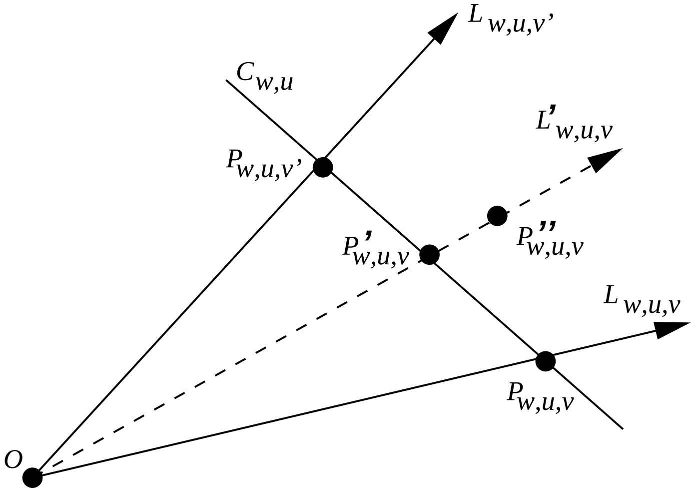

Figure 3: Points on the chord $C_{w,u}$.

For any $w \in \{0,1\}^+$, we define a dyadic rational number $x_w \in [0,1]$ as follows: Suppose $\ell(w) = n$, let i_w be the integer whose n-bit binary representation is equal to w, and let $x_w = 1 - 2^{-(n-1)} + i_w \cdot 2^{-2n}$. In addition, we let w' denote the successor of w in the lexicographic order. Note that the interval $[x_w, x_{w'}]$ has length $2^{-2\ell(w)}$.

Theorem 3.7 *Assume that* $A \in \Sigma_2^P$, *and satisfies* (3.1) *above. Let* $h_n = \cos(2^{-(p(n)+2n+2)}\pi)$ *for* $n \in \mathbb{N}$, *and for each* $w \in \{0,1\}^+$, *let* $\alpha_w = x_w\pi/2$. *Then, there exists a polynomial-time computable Jordan curve* Γ *such that, for all* $n \in \mathbb{N}$ *and* $w \in \{0,1\}^n$, *the following holds:*

(*a*) *If* $w \in A$ *then* $\min_{\alpha_w \le \alpha \le \alpha_{w'}} v(\alpha) = 2 + 2h_n$, *and*

(*b*) *If* $w \notin A$ *then* $\min_{\alpha_w \le \alpha \le \alpha_{w'}} v(\alpha) \ge 2 + 2h_n(1 + \delta_n)$,

where $v(t)$ *is the area of the circumscribed rectangle* R_t *of* Γ *at angle* t, *and* $\delta_n = 2^{-q(n)}$ *for some polynomial function* q.

Proof. The Jordan curve Γ is to be constructed from a unit circle Γ_0 with center $O = \langle 0,0\rangle$ and radius 1. The curve Γ is identical to Γ_0 on the second, third and fourth quadrants. That is, we will define a function $f : [0,1] \to \mathbb{R}^2$ to represent the curve Γ, and for $t \in [1/4, 1]$, $f(t) = \langle \cos(2t\pi), \sin(2t\pi)\rangle$.

Now we define f on $[0, 1/4]$. For any $n \in \mathbb{N}$ and $w \in \{0,1\}^n$, divide equally the interval $I_w = [x_w/4, x_{w'}/4]$ into $2^{p(n)}$ subintervals, with each one corresponding to a string $u \in \{0,1\}^{p(n)}$ (following the lexicographic order), denoted $I_{w,u}$. So the length of the interval $I_{w,u}$ is $2^{-(p(n)+2n+2)}$. Similarly, divide $I_{w,u}$ into $2^{p(n)}$ subintervals of equal length, with each one corresponding to a string $v \in \{0,1\}^{p(n)}$ and denoted $I_{w,u,v}$. So the length of an interval $I_{w,u,v}$ is $2^{-(2p(n)+2n+2)}$. Let $x_{w,u} = x_w/4 + i_u \cdot 2^{-(p(n)+2n+2)}$ and $x_{w,u,v} = x_{w,u} + i_v \cdot 2^{-(2p(n)+2n+2)}$, then $I_{w,u} = [x_{w,u}, x_{w,u'}]$ and $I_{w,u,v} = [x_{w,u,v}, x_{w,u,v'}]$. Let $P_{w,u}$ denote the point $\langle \cos(2x_{w,u}\pi), \sin(2x_{w,u}\pi)\rangle$. Let $C_{w,u}$ be the chord connecting the points $P_{w,u}$ and $P_{w,u'}$.

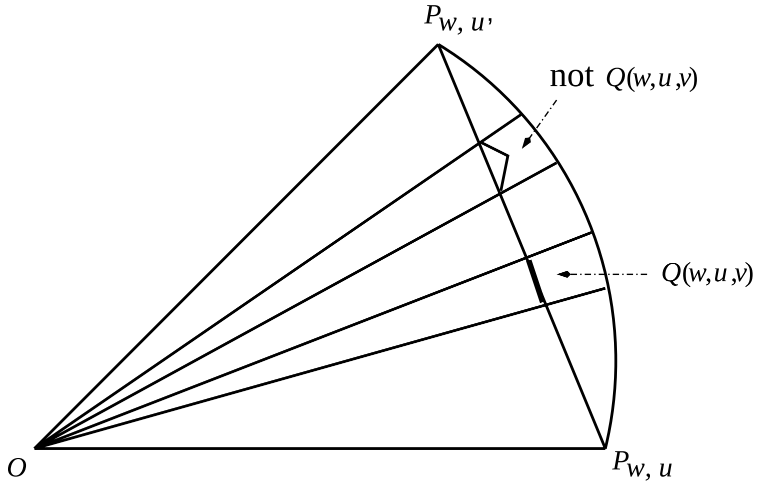

Figure 4: The function f on $I_{w,u}$

Also let $L_{w,u,v}$ be the half-line from the origin $\langle 0,0\rangle$ of angle $2x_{w,u,v}\pi$, and $L'_{w,u,v}$ the half-line from the origin of angle $(x_{w,u,v}+x_{w,u,v'})\pi$. Let $P_{w,u,v}$ denote the intersection point of the half-line $L_{w,u,v}$ and the chord $C_{w,u}$, and $P'_{w,u,v}$ the intersection point of the half-line $L'_{w,u,v}$ and the chord $C_{w,u}$. Finally, let $P''_{w,u,v}$ be the point on the half-line $L'_{w,u,v}$ with distance to the origin equal to the maximum of $length(\overline{OP_{w,u,v}})$ and $length(\overline{OP_{w,u,v'}})$ (see Figure 3).

We claim that the distance between $P'_{w.u.v}$ and $P''_{w,u,v}$ is at least $h_n \cdot 2^{-q(n)}$ for some polynomial function q. To see this, let us assume that $length(\overline{OP_{w,u,v}}) > length(\overline{OP_{w,u,v'}})$. Then, $length(\overline{OP''_{w,u,v}})$ equals $length(\overline{OP_{w,u,v}})$, which is at least $length(\overline{OP'_{w,u,v}})/\cos(2^{-(2p(n)+2n+2)}\pi)$. The claim follows now from the observation that $length(\overline{OP'_{w,u,v}}) > h_n$ and $1/\cos\alpha - 1 \geq \alpha^2/2$ when $\alpha \in [0,\pi/2)$.

Now, we define function f on $I_{w,u,v} = [x_{w,u,v}, x_{w,u,v'}]$ as follows (see Figure 4):

(1) If $Q(w,u,v)$,then f is linear on $I_{w,u,v}$ with $f(x_{w,u,v}) = P_{w,u,v}$ and $f(x_{w,u,v'}) = P_{w,u,v'}$.

(2) If $\neg Q(w,u,v)$, then f is piecewise linear on $I_{w,u,v}$ with three breakpoints $f(x_{w,u,v}) = P_{w,u,v}$, $f(x_{w,u,v'}) = P_{w,u,v'}$, and $f((x_{w,u,v}+x_{w,u,v'})/2) = \langle m\cos\beta, m\sin\beta\rangle = P''_{w,u,v}$, where $m = \max\{length(\overline{OP_{w,u,v}}), length(\overline{OP_{w,u,v'}})\}$, and $\beta = (x_{w,u,v}+x_{w,u,v'})\pi$.

It is not hard to see that f has a polynomial modulus function and is polynomial-time computable. We omit the details.

We now check conditions (a) and (b). First, we note that the bumps of Γ all lie within the unit circle Γ_0. Therefore, the circumscribed rectangle of Γ at an angle α in $[x_{w,u}, x_{w,u'}]$ must touch a point of Γ in this angle.

(a) If $w \in \{0,1\}^n \cap A$, then $(\exists u, \ell(u) = p(n))(\forall v, \ell(v) = p(n))Q(w,u,v)$. According to the definition of f, f on $I_{w,u}$ is a line segment with a distance of $\cos(2^{-(p(n)+2n+2)}\pi) = h_n$ from the origin (see Figure 4). Therefore, the minimum area $v(\alpha)$ for $\alpha \in [\alpha_w, \alpha_{w'}]$ is $2(1+h_n) = 2+2h_n$.

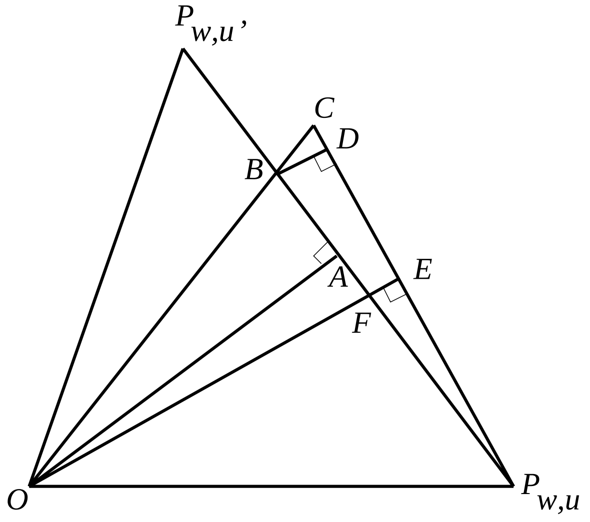

Figure 5: Proof of condition (b)

(b) If $w \in \{0,1\}^n - A$, then $(\forall u, \ell(u) = p(n))(\exists v, \ell(v) = p(n))\neg Q(w,u,v)$. Thus, on each $I_{w,u}$, f has at least a bump at $I_{w,u,v}$ (see Figure 4), with the tip of the bump of distance at least $h_n \cdot 2^{-q(n)}$ to $P'_{w,u,v}$. This implies that, for any line tangent to the portion of Γ between $P_{w,u}$ and $P_{w,u'}$, its distance to the origin is at least $h_n(1 + 2^{-(q(n)+2)})$. This can be seen from Figure 5: Let B denote $P'_{w,u,v}$ and C denote $P''_{w,u,v}$. Suppose the line segment $\overline{BC}$ has length ϵ, then $length(\overline{EF}) \geq 1/2 \cdot length(\overline{BD}) \geq 1/4 \cdot length(\overline{BC}) = \epsilon/4$. Therefore, the line $\overline{CP_{w,u}}$ has distance at least $length(\overline{OA}) + \epsilon/4 = h_n + \epsilon/4$ to the origin. $\square$

Corollary 3.8 *Assume that NP $\neq$ coNP. Then, there exists a polynomial-time computable Jordan curve* Γ*, such that the function* $v(a,b) = \min_{a \leq \alpha \leq b} v(\alpha)$ *is not computable by an NP oracle Turing machine, where* $v(\alpha)$ *is the area of the circumscribed rectangle of* Γ *at angle* α*.*

In the above construction, we embedded each question of whether $w \in A$ for a given $A \in \Sigma_2^P$ in a different angle of the curve Γ. It remains open whether we can embed them at a single angle. That is, the question of whether every right Σ_2^P-real number is equal to the minimum area of a circumscribed rectangle R_α of a polynomial-time computable Jordan curve Γ (without any restriction on the angle α) is left open.

4 Circumscribed Squares

It is not hard to see that the problem of computing the minimum area of a square enclosing a polynomial-time computable Jordan curve Γ is similar to that of a rectangle. Namely, we can guess the corners of a square and verify that they form a square and that every point of Γ is within the square. Therefore, the minimum area of an enclosing square of Γ is a right Σ_2^P-real number. In addition, with a construction that is only slightly different from that in the proof of Theorem 3.7, we can get results similar to Theorem 3.7 and Corollary 3.8.

It is important, however, to point out that this minimum square does not necessarily circumscribe the curve Γ. In fact, the mapping from a Jordan curve Γ to the area of its minimum enclosing square is a continuous function (with respect to the hausdorff distance between Jordan curves). However, the mapping from a curve Γ to its minimum circumscribed square is not a continuous function. This can be seen from the following simple example: The minimum circumscribed square of a square is itself, but the minimum circumscribed square of a rectangle of unequal sides is the square that forms a 45-degree angle with the rectangle (see Figure 6).

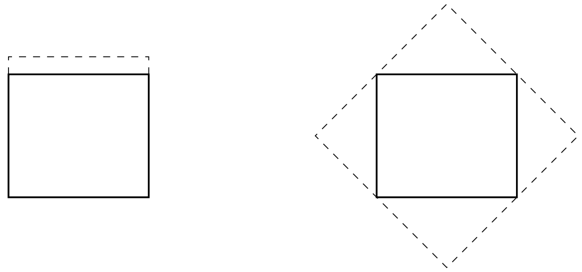

Figure 6: The minimum enclosing square and minimum circumscribed square of a rectangle of uneven sides

Thus, finding the minimum circumscribed square of a Jordan curve is a different question. Indeed, it is not immediately clear whether a Jordan curve must have a circumscribed square at all. It turns out that this question has an affirmative answer. Indeed, we can prove, by the intermediate value theorem that, for every Jordan curve Γ, there must exist at least one circumscribed square: For each $\alpha \in [0, \pi/2]$, let R_α denote the circumscribed rectangle of Γ at angle α, and let a be the length of the side of the angle α with the x-axis, and b the length of one of its neighboring side, and let $g(\alpha) = a - b$. Then, it is clear that g is continuous on $[0, \pi/2]$ and $g(0) = -g(\pi/2)$. So, by the intermediate value theorem, there exists an α in $[0, \pi/2)$ such that $g(\alpha) = 0$, and R_α is a circumscribed square of Γ.

It is well known that the intermediate value theorem has an effective proof. Namely, for any computable function $f : [0, 1] \to \mathbb{R}$ with $f(0) < 0 < f(1)$, there exists at least one computable point $x \in (0, 1)$ such that $f(x) = 0$.

Since the above function g is a computable function (actually, the difference of two *NP* real functions), it must have a computable root.

Proposition 4.1 *Every polynomial-time computable Jordan curve* Γ *on* $\mathbb{R}^2$ *has at least one computable circumscribed square.*

For the complexity of the minimum circumscribed square, we first note that the minimum circumscribed rectangles of the curve Γ constructed in the proof of Theorem 3.5 are actually all squares. Therefore, we see that a polynomial-time computable Jordan curve may not have a computable minimum circumscribed square.

On the other hand, we note that if the curve Γ has a unique circumscribed square then, by Proposition 4.1, it must be computable. In this case, what is the complexity of the square? We can answer this question again by way of the intermediate value theorem. We recall the following theorem about the complexity of the intermediate value theorem.

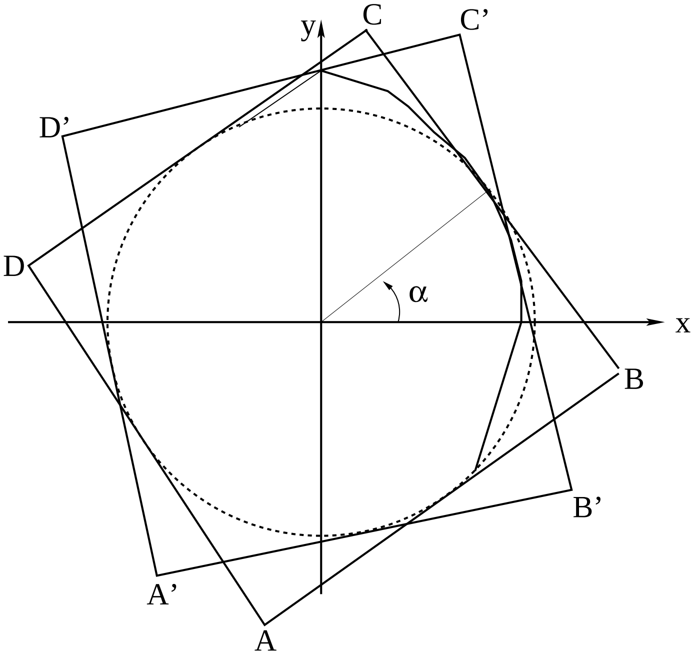

Figure 7: The function f

Proposition 4.2 ([10, *Theorem* 4.4]) *For any recursive real number $x \in [0,1]$, there exists a strictly increasing, polynomial-time computable function $h : [0,1] \to \mathbb{R}$ such that x is the unique root of h in $[0,1]$.*

A similar result holds for the unique circumscribed squares.

Theorem 4.3 *For any recursive real number $\alpha \in [0,\pi/2]$, there exists a a polynomial-time computable Jordan curve Γ such that its circumscribed rectangle R_α at angle α is its unique circumscribed square.*

Sketch of Proof. Without loss of generality, we assume that $\alpha \in (\pi/8, \pi/4)$. Let Γ_0 be the circle with center $\langle 0,0\rangle$ and radius 1. We construct the Jordan curve Γ from Γ_0 by shrinking the portion of Γ_0 in the first quadrant to the right of angle α inward, and enlarging the portion of Γ_0 in the first quadrant to the left of angle α (see Figure 7).

More precisely, we first construct, as in the proof of Proposition 4.2, a polynomial-time computable, piecewise linear function $h : [0,\pi/2] \to \mathbb{R}$ with the following properties:

(i) h is strictly increasing on $[0,\pi/2]$.

(ii) $|h(x)| \le 1$ for all $x \in [0,\pi/2]$.

(iii) α is the unique root of h on $[0,\pi/2]$.

(iv) $h(x) \le 1/\cos(x-\alpha) - 1$ for all $x \in [0,\pi/2]$, and $h(\pi/2) \le 1/\cos(\alpha) - 1$.

We then define a function $f : [0,2\pi] \to \mathbb{R}^2$ (as the representation of Γ) as follows:

(i) For $x \in [0, \pi/2]$, $f(x) = \langle (1 + h(x)) \cos x, (1 + h(x)) \sin x \rangle$ (i.e., on the first quadrant, Γ differs from Γ_0 by the amount of $h(x)$).

(ii) f is linear on $[\pi/2, 5\pi/8]$ with $f(\pi/2) = \langle 0, 1 + h(1) \rangle$ and $f(5\pi/8) = \langle \cos(5\pi/8), \sin(5\pi/8) \rangle$.

(iii) For $x \in [5\pi/8, 7\pi/4]$, $f(t) = \langle \cos x, \sin x \rangle$ (i.e., Γ is identical to Γ_0 on $[5\pi/8, 7\pi/4]$).

(iv) f is linear on $[7\pi/4, 2\pi]$ with $f(7\pi/4) = \langle \cos(7\pi/4), \sin(7\pi/4) \rangle$ and $f(2\pi) = \langle 1 + h(0), 0 \rangle$.

This design makes all circumscribed rectangles R_β of Γ at an angle $\beta < \alpha$ have negative $g(\beta)$ values, and those R_β with $\beta > \alpha$ have positive $g(\beta)$ values, where g is the difference between two neighboring sides of R_β as defined earlier. Note that property (iv) of function h ensures that R_α is a square. Thus, R_α is the unique circumscribed square of Γ. $\square$

We say a function $t(n)$ is a *fully-time constructible function* if there exists a Turing machine M that halts on input n in exactly $t(n)$ moves. Most familiar time bounds, such as n^2, 2^n, are fully-time constructible (see, e.g., Du and Ko [7]).

Corollary 4.4 *For any fully time-constructible function $t(n)$, there exists a polynomial-time computable Jordan curve Γ which has a unique circumscribed square S but S is not computable in time $t(n)$.*

5 Final Remarks

In this paper, we studied the computational complexity of finding, from a given polynomial-time computable Jordan curve, the circumscribed rectangles and squares of the minimum area. We applied the proof techniques for the general minimization problem to the minimum circumscribed rectangle problem, and showed results similar to the general minimization problem. In particular, we characterized the complexity of the area of the minimum circumscribed rectangle by the discrete complexity class Σ_2^P.

We note that, however, the known results about the general minimization problem cannot apply to our problem directly. Since we are dealing with geometric objects, the constructions have more constraints. In fact, for the minimum area of a circumscribed rectangle, there is a small gap between our upper bound (Theorem 3.6) and lower bound (Theorem 3.7). This is, as pointed out at the end of Section 3, because the extra constraints seem to interfere each other, we are not able to put the constructions around a single angle.

Furthermore, we remark that the results in Section 3 can be adapted to two other important concepts in computational geometry, namely, the minimum-perimeter circumscribed rectangles of a Jordan curve (see DePano[6] and Pirzadeh [13]) and the least width of a Jordan curve (see Houle and Toussaint [9] and Pirzadeh [13]).[3] We note that the perimeter and the width of a circumscribed rectangle R_α at a fixed angle α are, like the area of R_α,

---

[3] The width of a Jordan curve Γ at a given angle α is the distance of two parallel lines L_1 and L_2 at angle α such that every point of Γ is between L_1 and L_2 or on $L_1 \cup L_2$; the least and greatest widths are the minimum and maximum of widths over all angles, respectively.

left *NP*-real numbers. So, all the results from Corollary 3.4 to Corollary 3.8 also hold for these two concepts. It is interesting to point out that the maximum width of Γ, which is equal to the diameter of Γ,[4] is a left *NP*-real number, and hence has lower complexity than the least width of Γ, assuming that $NP \neq coNP$.

Finally, we discuss the differences between our results with those in computational geometry. Toussaint [1983] has shown that it takes $O(n)$ time to compute a minimum-area circumscribed rectangle of an n-sided polygon, while we have proved that the problem of finding minimum-area circumscribed rectangles of a polynomial-time computable Jordan curve is undecidable, and the problem of finding the minimum area of circumscribed rectangles of a polynomial-time computable Jordan curve is in Σ_2^P. This difference stems from the different computational models and complexity measures used in the two approaches. In the computational geometry approach, the curves to be studied are restricted to be polygons, and the input n-sided polygons are presented to the algorithm with the n vertices given explicitly. In addition, the time complexity of the algorithm is measured against the size n of the input polygon. In our approach, the algorithm needs to work on all polynomial-time computable curves, not just polygons, and the time complexity is measured with respect to the output precision of the circumscribed rectangle. We note that although the curve Γ in our model may be approximated by polygons, an approximate polygon with error $\leq 2^{-n}$ would have $2^{p(n)}$ vertices for some polynomial p. When it is applied to this approximate polygon, Toussaint's algorithm would take exponential time (with respect to the precision n). In other words, our approach considers a wider range of problems, and the results are consistent with the results from computational geometry.

References

[1] M. Braverman, Hyperbolic Julia sets are poly-time computable, in: Proceedings of the 6th Workshop on Computability and Complexity in Analysis (CCA 2004), Electr. Notes Theor. Comput. Sci. 120 (2005) 17–30.

[2] M. Braverman and M. Yampolsky, Non-computable Julia sets, CoRR math.DS/0406416 (http://arxiv.org/abs/math/0406416).

[3] A. W. Chou and K.-I. Ko, Computational Complexity of Two-Dimensional Regions, SIAM J. Comput. 24 (1995) 923–947.

[4] A. W. Chou and K.-I. Ko, On the complexity of finding paths in a two-dimensional domain I: Shortest paths, Math. Log. Q. 50-6 (2004) 551–572.

[5] A. W. Chou and K.-I. Ko, On the complexity of finding paths in a two-dimensional domain II: Piecewise straight-line paths, in: Proceedings of the 6th Workshop on Computability and Complexity in Analysis (CCA 2004), Electr. Notes Theor. Comput. Sci. 120 (2005) 45–57.

[6] N. A. A. DePano, Rotating calipers revisited: optimal polygonal enclosures in optimal time, in: Proceedings of ACM South Central Regional Conference, Lafayette, LA, November 1987.

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

---

[4] The diameter of Γ is the maximum length of line segments $\overline{AB}$ over all pairs $A, B \in \Gamma$.

[8] H. Freeman and R. Shapira, Determining the minimum-area encasing rectangle for an arbitrary closed curve, Commun. ACM 18-7 (1975) 409–413.

[9] M. E. Houle and G. T. Toussaint, Computing the width of a set, IEEE Trans. Pattern Anal. Mach. Intell. 10-5 (1988) 761–765.

[10] K.-I. Ko, Complexity Theory of Real Functions, Birkh*ä*user, Boston, 1991.

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

[12] K.-I. Ko and F. Yu, On the complexity of computing the logarithm and square root functions on a complex domain, in: Proceedings of the 11th International Computing and Combinatorics Conference (COCOON 2005), Lecture Notes in Comput. Sci. 3595 (2005) 349-358.

[13] H. Pirzadeh, Computational geometry with the rotating calipers, Master's thesis, McGill University, 1999.

[14] M. B. Pour-El and J. I. Richards, Computability in Analysis and Physics, Springer, Berlin, 1989.

[15] R. Rettinger, A fast algorithm for Julia sets of hyperbolic rational functions, in: Proceedings of the 6th Workshop on Computability and Complexity in Analysis(CCA 2004), Electr. Notes Theor. Comput. Sci. 120 (2005) 145–157.

[16] R. Rettinger and K. Weihrauch, The computational complexity of some Julia sets, in: Proceedings of the 35th Annual ACM Symposium on Theory of Computing(STOC 2003), pp. 177–185, San Diego, CA, 2003.

[17] P. J. Schneider and D. Eberly, Geometric Tools for Computer Graphics, Elsevier, New York, 2003.

[18] E. Specker, Der satz vom maximum in der rekursiven analysis, in: Constructivity in Mathematics, pp. 254–265, North-Holland, 1959.

[19] G. T. Toussaint, Solving geometric problems with the rotating calipers, in: Proc. MELECON '83, Athens, 1983.

[20] K. Weihrauch, Computable Analysis, Springer, Berlin, 2000.