

FROM CURVES TO WORDS AND BACK AGAIN:
GEOMETRIC COMPUTATION OF MINIMUM-AREA HOMOTOPY

by

Bradley Allen McCoy

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

Doctor of Philosophy

in

Computer Science

MONTANA STATE UNIVERSITY
Bozeman, Montana

May 2024

©COPYRIGHT

by

Bradley Allen McCoy

2024

All Rights Reserved

DEDICATION

For Dorothy.

ACKNOWLEDGEMENTS

I would like to thank all of my teachers for their encouragement and support. Of special note are Drs. Brittany Fasy and David Millman, who encourage me to explore, ask for help, and write. I would also like to thank Drs. Binhai Zhu, and Ryan Grady for answering my questions and serving on my committee. Many others have had a positive influence on my technical development, a subset includes Adam H., Adam W., Adisha, Andy, Anna C., Anna S., Ben, Carola, Corrine, Dan, David A., Eli, Hsien-Chih, Jordan, Lucy, Sam, Stacey, Stephan, and Stijn. I thank my parents for their unconditional love and my sister Tracy for being the ideal role model. I am deeply grateful to Anna, Dorothy, Gemma, and Murphy for being my family. I also thank the National Science Foundation for the financial support. This work is supported by NSF grants DMS 1664858 and CCF 2046730.

TABLE OF CONTENTS

1. INTRODUCTION	1
1.1 Our Contributions	2
2. BACKGROUND.....	4
2.1 Curves and Graphs	4
2.2 Homotopy and Isotopy	5
2.3 Self-Overlapping Curves	8
3. FROM CURVES TO WORDS.....	11
3.1 Blank’s Word Construction	12
3.2 The Nie Word Construction	17
3.3 Word Equivalence	19
4. FOLDINGS AND SELF-OVERLAPPING DECOMPOSITIONS	23
4.1 The Cancellation Norm and Blank Cuts.....	23
4.2 Compute Min-Area Homotopy from Self-Overlapping Decomposition	36
4.3 Min-Area Self-Overlapping Decomposition in Polynomial Time	38
5. AN ALGORITHM	41
5.1 A Polynomial Implementation	41
5.2 Curve Similarity	44
5.3 Fréchet Distance.....	46
5.4 Experiments	47
5.5 Discussion	48
6. HOMOTOPY AREA ON SURFACES WITH POSITIVE GENUS.....	54
6.1 The Universal Cover	55
6.2 An Algorithm for Contractible Curves on a Surface	57
6.3 Connector Between Homotopic Closed Curves	61
6.4 Putting the Pieces Together	64
6.5 Discussion	66
7. FROM WORDS TO CURVES.....	69
7.1 A Special Case	70
7.2 Combined Words to Curves	73

TABLE OF CONTENTS – CONTINUED

7.3 Blank Words to Curves	82
REFERENCES CITED.....	83

LIST OF TABLES

Table		Page
5.1	The coordinates for the paths shown in Figure 5.4.....	50
5.2	The time, in seconds, to compute the Fréchet distance and the minimum homotopy area for each pair of paths given above.	51

LIST OF FIGURES

Figure	Page
2.1 A generic plane curve induces a four-regular graph.	4
2.2 An example of a minimum area homotopy is shown. (a) A generic closed curve in the plane. (b) We see a homotopy that sweeps over the face f_3 . (c) The homotopy sweeps f_3 again. (d) The homotopy avoids sweeping over the face f_2 . This is a minimum area homotopy for the curve, the area is $\text{Area}(f_1) + 2 \cdot \text{Area}(f_3)$	7
2.3 The three homotopy moves are illustrated. (a) A $0 \leftrightarrow 1$ move removes or creates a empty loop. (b) A $0 \leftrightarrow 2$ move removes or creates a bigon. (c) A $3 \leftrightarrow 3$ move flips an empty triangle by moving one subpath over the opposite intersection point.	8
2.4 Examples and non-examples of self-overlapping decompositions. (a) Curve γ with intersection sequence $\gamma_V = [v_0, v_1, v_1, v_2, v_2, v_0]$. (b) All vertices are paired. (c) One of the subcurves is not self-overlapping. (d) Both subcurves are self-overlapping.	10
3.1 An example illustrating the choices made when constructing the Blank word. (a) A curve γ with labeled faces and edges, Π_a is drawn in blue. The Blank word of γ corresponding to Π_a is $[\gamma]_B(\Pi_a) = [23142\bar{3}4]$. (b) The same curve with a different choice of cables Π_b . The corresponding Blank word is $[\gamma]_B(\Pi_b) = [3214\bar{3}2\bar{4}]$	12
3.2 An example illustrating how to manage a cable system. (a) A cable system Π_1 on γ that is not managed. The red cables do not follow existing paths to the exterior face. (b) A managed cable system Π_2 on γ . (c) The dual γ^* in red. (d) The spanning tree T^* in γ^* generated by the managed cable system Π_2	14
3.3 An example of a cable system with a cable, in red, that cannot be managed.	16

LIST OF FIGURES – CONTINUED

Figure	Page
3.4 (a) A curve with labeled faces and edges. A shortest path cable system Π_a is drawn in blue. From the indicated start position, the Blank Word is $[\gamma]_B(\Pi_a) = [43254\bar{3}1]$. (b) The same curve with a cable system Π_b that does not fulfill the shortest path assumption. The Blank word is $[\gamma]_B(\Pi_b) = [43254\bar{2}\bar{3}21]$	16
3.5 As we traverse the red edge γ_r intersects 3, 2 , then 1 —the cable corresponding to f_1 —then 4 . As we traverse the boundary of f_1 we traverse the red edge, followed by $\bar{4}$ and $\bar{2}\bar{3}$. This choice of cable system Π corresponds to cycle flattening at f_1 as $\bar{2}\bar{3}\bar{3}21\bar{4}\bar{4}$ by writing $f_1 = \bar{e}_2 e_3 e_1 e_4$, or equivalently, $\bar{2}\bar{3}\bar{3}21\bar{4}\bar{4} = \bar{e}_2 \bar{e}_3 e_1 e_4$. The Blank subword on γ_r , with respect to the cycle flattening, is $e_1 = \bar{3}\bar{2} \cdot \bar{2}\bar{3}\bar{3}21\bar{4}\bar{4} \cdot \bar{4} = 3214$, as expected.	21
3.6 An example illustrating the choices made when breaking a cycle. (a) A curve γ . (b) A spanning tree in red and cotree in blue. (c) A labeling of the coedges and faces. We have four ways to break the cyclic face sequence for e_1 , represented using the cable system Π : (d) $e_1 = \partial f_1 e_2 \bar{e}_3 \bar{e}_4$; (e) $e_1 = e_2 \partial f_1 \bar{e}_3 \bar{e}_4$; (f) $e_1 = e_2 \bar{e}_3 \partial f_1 \bar{e}_4$; and (g) $e_1 = e_2 \bar{e}_3 \bar{e}_4 \partial f_1$	22
4.1 An example of a Blank cut. (a) A curve with labeled path P . (b) The two induced subcurves from cutting along P	25
4.2 An example illustrating how to avoid partially cutting a face. (a) A curve with cables. (b) Isotopy the cables to not partially cut any faces. (c) One subcurve resulting from cutting along the middle cable. The curve is weakly simple and there are two cables in this face. (d) The other subcurve.	25

LIST OF FIGURES – CONTINUED

Figure	Page
4.3 An example of a curve with two extensions that are not diffeomorphic. (a) Milnor’s curve. A possible word corresponding to this curve is $w = [251346\bar{5}3467253\bar{6}]$. (b) An extension corresponding to the grouping $[251346\bar{5}3467253\bar{6}]$. The blue and green symbols represent pairings. (c) An extension corresponding to the grouping $[251346\bar{5}3467253\bar{6}]$	26
4.4 An illustration of a Dehn twist.	31
4.5 (a) An example of the Dehn twist. The curve is shown in black. Starting from the star and following the indicated orientation, the Blank word is $w = [23142\bar{3}4]$. In the instance we have $i = 4$ and $j = 3$; the curve $c_{4,3}$ is shown in gray. The word B is 1 . (b) After twisting about an annulus formed by fattening $c_{4,3}$ the new word is $w' = [2141\bar{3}3314\bar{1}141314\bar{1}31421413\bar{3}314141314\bar{1}314]$	33
4.6 A curve with combined word $[c4c4231d\bar{5}da2b\bar{3}ba]$	36
4.7 We decompose the curve in (a) at vertex v into self-overlapping subcurves, the cable system on the induced subcurve in (b) has more than one marked point in a face and cables do not follow shortest paths.	37
4.8 (a) A k -stack where $k = 2$. Smoothing at any of the three vertices gives a self-overlapping decomposition. (b) The vertex v is a sign-changing vertex.	39
5.1 An example of a curve with a Blank word $\Omega(F ^2)$. (a) A curve with Blank word $[1\bar{2}345]$. (b) Each time we wrap the curve, each of the five faces appears in the word again. The word is now $[1\bar{2}345123456]$. (c) Wrapping again makes the word $[1\bar{2}3451234561234567]$. (d) Wrapping one more time makes the word $[1\bar{2}345123456123456712345678]$. There are $\frac{ F }{2}$ faces each with depth $\frac{ F }{2}$ and the length of the word is $\Omega(F ^2)$	42

LIST OF FIGURES – CONTINUED

Figure	Page
5.2 An illustration of two paths and their corresponding free space diagram. (a) A person and their dog realizing the Fréchet distance between two paths. (b) The free space between the two paths. The free space diagram was generated using the applet of Schäfer [63].	46
5.3 An example showing the free space of two paths. (a) Two paths that each consist of a single segment. (b) The free space between the two segments is convex. The free space diagram was generated using the applet of Schäfer [63].	48
5.4 Four pairs of paths are shown. The homotopy area is of the closed curve formed by connecting the endpoints of the paths. (a) Paths P and Q with Fréchet distance 2.235 and minimum homotopy area 8.5. (b) Paths P and Q with Fréchet distance is 2.235 and minimum homotopy area is 7.333. (c) Paths P and Q with Fréchet distance 4.123 and minimum homotopy area is 4.00455. (d) Paths P and Q with Fréchet distance 10.050 and minimum homotopy area is 72.0663.	49
5.5 A trajectory illustrating an error that significantly changes the Fréchet distance but only slightly changes the homotopy area. Permission to include image granted by [44].	51
5.6 Two curves on a polygonal terrain with relatively small Fréchet distance in \mathbb{R}^3 and a relatively large geodesic Fréchet distance.	52
6.1 We see three curves on a surface of genus two. Two of the three curves are homotopic. Algorithm 6.2 computes the minimum homotopy area between them. The image of a minimum area homotopy is shaded.	55
6.2 (a) A genus two surface with a system of loops. (b) The polygonal schema corresponding to the surface and the system of loops.	57
6.3 Two curves that are not homotopic when the basepoint is fixed but they are freely homotopic.	58

LIST OF FIGURES – CONTINUED

Figure	Page
6.4 An example illustrating that care must be taken in how we connect the curves. (a) Two homotopic curves on a surface connected by a path. The resulting curve is in the homotopy class shown in (b) and is not contractible.....	62
6.5 An illustration of a null-homotopy of a curve of the form $\gamma_1 \circ \beta \circ \text{rev}(\gamma_2) \circ \text{rev}(\beta)$. The area of the homotopy is equal to the area of the homotopy in Figure 6.6. (a) A closed curve C drawn in blue on a surface. In Figure 6.6, we construct a homotopy from γ_1 to γ_2 with equal area. (b) A null-homotopy of the curve in progress. (c) The homotopy continues. (d) The null-homotopy ends at a point.	64
6.6 An illustration of a homotopy between two curves that has equal area as the homotopy of the curve in Figure 6.5. (a) Two homotopic curves γ_1 and γ_2 on a surface. For any homotopy of C we construct a homotopy from γ_1 to γ_2 that sweeps out equal area. (b) We add a spike from γ_1 that follows the connector of C then follows γ_2 . (c) We now contract the copy of C using the homotopy of C in Figure 6.5. (d) We continue contracting. (e) We are left with a copy of γ_2 with an addisitional spike. Thus, we have a homotopy form γ_1 to γ_2 with area equal to the area of C	65
7.1 Some Blank words correspond to curves and others do not. (a) There does exist a curve with Blank word [122]. (b) There does not exist a curve with Blank word [12].	70
7.2 Some Gauss codes correspond to curves and others do not. (a) There does exist a curve with Gauss code [aabb]. (b) There does not exist a curve with Gauss code [abab].	70
7.3 In linear time, we add pointers from each vertex to the other occurrence of the vertex. In (a) we have a curve with labeled vertices and in (b) the Gauss diagram for the curve.	71
7.4 A curve where each face has depth one. There are four monogons.	71

LIST OF FIGURES – CONTINUED

Figure	Page
<p>7.5 We are given the Blank word $w = [1\bar{2}\bar{3}4\bar{5}6]$. Since w has positive and negative letters, we construct a curve that corresponds to w. (a) A curve that corresponds to two letters with opposite sign, $[2\bar{3}]$. (b) We add the next letter, $[2\bar{3}4]$. It is positive, so we add it to the negative part of the attachment segment. (c) We add the next letter, $[2\bar{3}4\bar{5}]$. It is negative, so we add it to the positive part of the attachment segment. (d) The next two letters are positive, $[2\bar{3}4\bar{5}61]$. So we add them to the negative part of the attachment segment and we have constructed a curve that corresponds to w.</p>	73
<p>7.6 The crossing from right to left is positive and the crossing from left to right is negative.</p>	74
<p>7.7 An example of tracing out a face of depth one. (a) A curve with combined word $w = [t312hba32h54ta3b4]$. We wish to find the boundary of the face with label one. The initial edge, and root of our tree, is $t312h$. (b) The vertices adjacent to <code>match[t]</code> give the substrings a and $h54$, we prepend them to the current string to become the children of the root. Since we reach h, we make a leaf, we then check if face one is the only face that appears exactly once. It is not, so we have not found the boundary. We then consider <code>match[a]</code>. We find h and this time, face one is the only face that appears exactly once, and we have found the boundary of face one.</p>	78

LIST OF ALGORITHMS

Algorithm	Page
5.1 PolynomialMinimumAreaHomotopy(γ)	43
6.2 MinimumSurfaceHomotopy(S, γ_1, γ_2)	67
7.3 FaceTrace(w, f, e)	79
7.4 DrillDown(w, f, e)	80

ABSTRACT

Let γ be a generic closed curve in the plane. The area of a homotopy is the area swept by the homotopy. We consider the problem of computing the minimum null-homotopy area of γ . Samuel Blank, in his 1967 Ph.D. thesis, determined if γ is self-overlapping by *geometrically* constructing a combinatorial word from γ . More recently, Zipei Nie, in an unpublished manuscript, computed the minimum homotopy area of γ by constructing a combinatorial word *algebraically*. We provide a unified framework for working with both words and determine the settings under which Blank's word and Nie's word are equivalent. Using this equivalence, we give a new geometric proof for the correctness of Nie's algorithm. Unlike previous work, our proof is constructive which allows us to naturally compute the actual homotopy that realizes the minimum area. Furthermore, we contribute to the theory of self-overlapping curves by providing the first polynomial-time algorithm to compute a self-overlapping decomposition of any closed curve γ with minimum area.

Next, we describe the first polynomial implementation of an algorithm to compute the minimum homotopy area of a piecewise linear closed curve in the plane. We discuss how minimum homotopy area can be used as a similarity measure for curves and include experiments that compare the runtime of our algorithm to an implementation of the Fréchet distance. We then extend our algorithm for computing the minimum homotopy area in the plane to homotopic, non-intersecting, non-contractible curves on an orientable surface with positive genus. Finally, we consider the inverse problem of determining which combinatorial Blank words correspond to closed curves in the plane. We solve a special case of this problem and give an exponential algorithm to the general case.

CHAPTER ONE

INTRODUCTION

A *closed curve* in the plane is a continuous map γ from the circle \mathbb{S}^1 to the plane \mathbb{R}^2 . In the plane, any closed curve is homotopic to a point. A homotopy that sweeps out the minimum possible area is a *minimum homotopy*. Chambers and Wang [16] introduced the minimum homotopy area between two simple homotopic paths with common endpoints as a way to measure the similarity between the two curves. They suggest that homotopy area is more robust against noise than another popular similarity measure on curves called the Fréchet distance and that homotopy area can be naturally adapted to measure curve similarity on other surfaces. However, their algorithm requires that each path be simple, which is restrictive.

Fasy, Karakoç, and Wenk [39] proved that the problem of finding the minimum homotopy area is easy on a closed curve that is the boundary of an immersed disk. Such curves are called *self-overlapping* [37, 49, 53, 64, 67, 71]. They also established a tight connection between minimum-area homotopy and self-overlapping curves by showing that any generic closed curve can be decomposed at some vertices into self-overlapping subcurves such that the combined homotopy from the subcurves is minimum. This structural result gives an exponential-time algorithm for the minimum homotopy area problem by testing each decomposition in a brute-force manner.

Nie, in an unpublished manuscript [56], described a polynomial-time algorithm to determine the minimum homotopy area of any closed curve in the plane. Nie's algorithm borrows tools from geometric group theory by representing the curve as a word in the fundamental group $\pi_1(\gamma)$, and connects minimum homotopy area to the cancellation

norms [8, 10, 58] of the word, which can be computed using a dynamic program. However, the algorithm does not naturally compute an associated *minimum-area homotopy*.

Alternatively, one can interpret the words from the dynamic program *geometrically* as crossing sequences by traversing any subcurve cyclicly and recording the crossings along with their directions with a collection of nicely-drawn *cables* from each face to a point at infinity. Such geometric representation is known as the Blank words [6, 59]. In fact, the first application of these combinatorial words given by Blank is an algorithm that determines if a curve is self-overlapping. Blank words are geometric in nature and thus the associated objects are polynomial in size. When attempting to interpret Nie's dynamic program from the geometric view, one encounters the question of how to extend Blank's definition of cables to *subcurves*, where the cables inherited from the original curve are no longer positioned well with respect to the subcurves. To our knowledge, no geometric interpretation of the dynamic program is known.

1.1 Our Contributions

We first show that Blank and Nie's word constructions are, in fact, equivalent under the right assumptions (Chapter 3). Next, we extend the definition of Blank's word to subcurves and arbitrary cable drawings, and interpret the dynamic program by Nie geometrically (Chapter 4). Using the self-overlapping decomposition theorem by Fasy, Karakoç, and Wenk [39] we provide a correctness proof to the algorithm. We then show that a minimum-area self-overlapping decomposition can be found in polynomial time. We emphasize that extending Blank words to allow arbitrary cables is in no way straightforward. In fact, many assumptions on the cables have to be made in order to connect self-overlapping curves and minimum-area homotopy; handling arbitrary cable systems, as seen in the dynamic program, requires further tools from geometric topology like *Dehn twists*. Chapters 3 and 4 are an extension of [19] which is in collaboration with Hsien-Chih Chang, Brittany Terese Fasy,

David L. Millman, and Carola Wenk.

Next, we describe the first polynomial implementation of an algorithm to compute the minimum homotopy area in polynomial time (Chapter 5). We include experiments that show, in many cases, we can compute the minimum homotopy area faster than the Fréchet distance.

We then consider computing the minimum homotopy area between two non-contractible homotopic curves on an orientable surface with positive genus (Chapter 6). We show that we can apply our algorithm for optimal homotopies in the plane to other surfaces by lifting the curves to the universal cover of the surface.

Finally, we consider the problem of determining which Blank words correspond to closed curves in the plane (Chapter 7). This question is closely related to the historic problem of determining which Gauss codes correspond to closed curves. If the Blank word and the Gauss code are combined, we give a simple polynomial algorithm to determine if the sequence corresponds to a curve. Our results lead to an exponential algorithm to determine if a Blank word corresponds to a closed curve. We also give a polynomial algorithm to solve a special case of this problem where each letter in the word appears exactly once.

CHAPTER TWO

BACKGROUND

In this chapter, we introduce concepts and definitions that are used throughout the document. We assume the readers are familiar with the basic terminology for curves and surfaces.

2.1 Curves and Graphs

A *closed curve* in the plane is a continuous map $\gamma : \mathbb{S}^1 \rightarrow \mathbb{R}^2$, and a *path* in the plane is a continuous map $\zeta : [0, 1] \rightarrow \mathbb{R}^2$. A path ζ is *closed* when $\zeta(0) = \zeta(1)$. In this work, we are presented with a *generic* curve; that is, one where there are a finite number of self-intersections, each of which is transverse and no three strands cross at the same point. In other words, there are exactly two subpaths of the curve at each self-crossing, and the corresponding two tangent vectors must be linearly independent. See Figure 2.1 for an example. The image of a generic closed curve is naturally associated with a four-regular

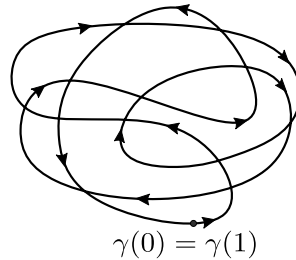


Figure 2.1: A generic plane curve induces a four-regular graph.

plane graph. The self-intersection points of a curve are *vertices*, the paths between vertices are *edges*, and the connected components of the complement of the curve are *faces*. Given a curve, choose an arbitrary starting point $\gamma(0) = \gamma(1)$ and orientation for γ .

The *dual graph* γ^* is another (multi-)graph, whose vertices represent the faces of γ , and two vertices in γ^* are joined by an edge if there is an edge between the two corresponding faces in γ . The dual graph is another plane graph with an inherited embedding from γ .

Let T be a spanning tree of γ . Let E denote the set of edges in γ , the tree T partitions E into two subsets, T and $T^* := E \setminus T$. The edges in T^* define a spanning tree of γ^* called the *cotree*. The partition of the edges (T, T^*) is called the *tree-cotree pair*.

We call a rooted spanning cotree T^* of γ^* a *breadth-first search tree* (BFS-tree) if it can be generated from a breadth-first search rooted at the vertex in γ^* corresponding to the unbounded face in γ . Each bounded face f of γ is a vertex in a breadth-first search tree T^* , we associate f with the unique edge incident to f^* in the direction of the root. Thus, there is a correspondence between edges of T^* and faces of γ .

2.2 Homotopy and Isotopy

A *homotopy* between two closed curves γ_1 and γ_2 that share a point p_0 is a continuous map $H: [0, 1] \times \mathbb{S}^1 \rightarrow \mathbb{R}^2$ such that $H(0, \cdot) = \gamma_1$, $H(1, \cdot) = \gamma_2$, and $H(s, 0) = p_0 = H(s, 1)$. We define a homotopy between two paths similarly, where the two endpoints are fixed throughout the continuous morph. Notice that homotopy between two closed curves as *closed curves* and the homotopy between them as *closed paths* with an identical starting points are different. A homotopy between two injective paths ζ_1 and ζ_2 is an *isotopy* if every intermediate path $H(s, \cdot)$ is injective for all s . The notion of isotopy naturally extends to a collection of paths.

Let X denote any topological space, and let $\xi: \mathbb{S}^1 \rightarrow X$ be a closed curve in X . We choose a basepoint $p \in \mathbb{S}^1$, and a point $q \in X$ and consider the set of basepoint preserving cycles through q with $\xi(p) = q$. The product of two cycles through q is defined as the concatenation of the cycles. The inverse ξ^{-1} of a cycle ξ through q is found by reversing the orientation of ξ . Homotopies between cycles that fix p is an equivalence relation and

the equivalence classes of cycles through q forms a group under the product and inverse operations described above. This group is the *fundamental group* of X at q and is denoted $\pi_1(X, q)$. All topological spaces we consider are path-connected, and the fundamental groups at all points are isomorphic so we write $\pi_1(X)$.

For $\gamma : \mathbb{S}^1 \rightarrow \mathbb{R}^2$, we can think of the image of γ in the plane as a topological space and consider $\pi_1(\gamma)$. The fundamental group of γ is a free group with basis consisting of the classes corresponding to the cotree edges of any tree-cotree pair of γ . Elements of the fundamental group are usually called words, whose letters correspond to equivalence classes of homotopic closed paths in γ . However, we will define a word to be a different object in Chapter 3, so we specify when we refer to an element of a fundamental group to avoid ambiguity.

Let H be a homotopy between curves γ_1 and γ_2 . Let $\#H^{-1}(x) : \mathbb{R}^2 \rightarrow \mathbb{Z}$ be the function that assigns to each $x \in \mathbb{R}^2$ the number of times the intermediate curves H sweep over x . The homotopy area of H is

$$\text{Area}(H) := \int_{\mathbb{R}^2} \#H^{-1}(x) dx.$$

The minimum area homotopy between γ_1 and γ_2 is the infimum of the homotopy area over all homotopies between γ_1 and γ_2 . We denote this by

$$\text{Area}_H(\gamma_1, \gamma_2) := \inf_H \text{Area}(H).$$

When γ_2 is the constant curve at a specific point p_0 on γ_1 , define

$$\text{Area}_H(\gamma) := \text{Area}_H(\gamma, p_0).$$

See Figure 2.2 for an example of a homotopy.

The existence of a homotopy with minimum area is due to the theory of minimal

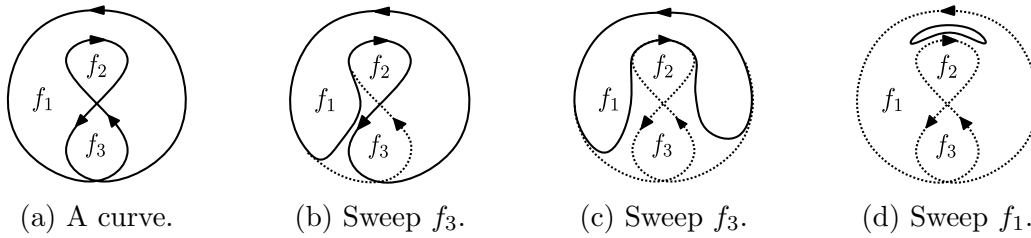


Figure 2.2: An example of a minimum area homotopy is shown. (a) A generic closed curve in the plane. (b) We see a homotopy that sweeps over the face f_3 . (c) The homotopy sweeps f_3 again. (d) The homotopy avoids sweeping over the face f_2 . This is a minimum area homotopy for the curve, the area is $\text{Area}(f_1) + 2 \cdot \text{Area}(f_3)$.

surfaces. Plateau’s problem asks: given a boundary, does there exist a minimal surface with that boundary? Rado and Douglas proved the existence of such a surface [28, 60].

For each $x \in \mathbb{R} \setminus \gamma$, the *winding number* of γ at x , denoted as $\text{wind}(x, \gamma)$, is the number of times γ “wraps around” x , with a *positive* sign if it is counterclockwise, and *negative* sign otherwise. The winding number is a constant on each face. The *winding area* of γ is defined to be the integral

$$\text{Area}_w(\gamma) := \int_{\mathbb{R}^2} |\text{wind}(x, \gamma)| dx = \sum_{\text{face } f} |\text{wind}(f, \gamma)| \cdot \text{Area}(f).$$

The *depth* of a face f is the minimal number of edges crossed by a path from f to the exterior face. The depth is a constant on each face. We say the depth of a curve is equal the maximum depth over all faces. We define the *depth area* to be

$$\text{Area}_D(\gamma) := \int_{\mathbb{R}^2} \text{depth}(x, \gamma) dx = \sum_{\text{face } f} \text{depth}(f) \cdot \text{Area}(f).$$

Chambers and Wang [16] showed that the winding area gives a lower bound for the minimum homotopy area. On the other hand, there is always a homotopy with area $\text{Area}_D(\gamma)$; one such homotopy can be constructed by smoothing the curve at each

vertex into simple depth cycles [18], then contracting each simple cycle. Therefore, we have

$$\text{Area}_W(\gamma) \leq \text{Area}_H(\gamma) \leq \text{Area}_D(\gamma). \quad (2.1)$$

When a curve is transformed via homotopy, we necessarily have intermediate curves that are not generic. A curve is *almost generic* if it has a finite number of self-intersections that are either triple points or non-transverse. A homotopy is a *generic homotopy* if each intermediate curve is piecewise generic, or almost generic with a finite number of intermediate curves that are almost generic. Any generic homotopy can be described by a sequence of homotopy moves. These moves are similar to Reidemeister moves in knot theory. There are three types of moves as shown in Figure 2.3. For a generic curve with $|V|$ self-intersections, Chang and Erickson show that $\Theta(|V|^{3/2})$ are sometimes necessary and always sufficient to simplify the curve to a simple curve [18].

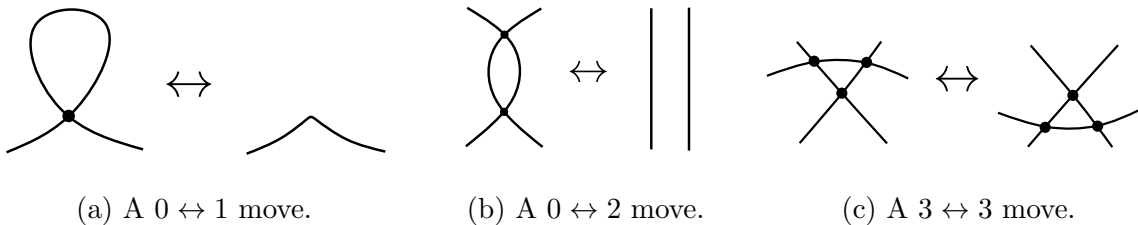


Figure 2.3: The three homotopy moves are illustrated. (a) A $0 \leftrightarrow 1$ move removes or creates a empty loop. (b) A $0 \leftrightarrow 2$ move removes or creates a bigon. (c) A $3 \leftrightarrow 3$ move flips an empty triangle by moving one subpath over the opposite intersection point.

2.3 Self-Overlapping Curves

A generic curve γ is *self-overlapping* if there exists an immersion of the disk $F : \mathbb{D}^2 \rightarrow \mathbb{R}^2$ such that $\gamma = F|_{\partial\mathbb{D}^2}$. We say a map F *extends* γ . The image $F(\mathbb{D}^2)$ is the *interior* of γ . There are several equivalent ways to define self-overlapping curves [37, 49, 53, 64, 67]. Properties

of self-overlapping curves are well-studied [36]; in particular, any self-overlapping curve has rotation number 1, where the *rotation number* of a curve γ is the winding number of the derivative γ' about the origin [71]. Also, the minimum homotopy area of any self-overlapping curve is equal to its winding area: $\text{Area}_W(\gamma) = \text{Area}_H(\gamma)$ [39].

The study of self-overlapping curves traces back to Whitney [71] and Titus [67]. Polynomial-time algorithms for determining if a curve is self-overlapping have been given [6, 64], as well as NP-hardness result for extensions to surfaces and higher-dimensional spaces [34]. See the thesis by Evans [36] for a more detailed history.

For any curve, the *intersection sequence*¹ $[\gamma]_V$ is a cyclic sequence of vertices $[v_0, v_1, \dots, v_{n-1}]$ with $v_n = v_0$, where each v_i is an intersection point of γ . Each vertex appears exactly twice in γ_V . Two vertices x and y are *linked* if the two appearances of x and y in γ_V alternate in cyclic order: $\dots x \dots y \dots x \dots y \dots$.

A pair of symbols of the same vertex x induces two natural subcurves generated by *smoothing* the vertex x ; see Figure 2.4 for an example. (In this work, every smoothing is done in the way that respects the orientation and splits the curve into two subcurves.) A *vertex pairing* is a collection of pairwise unlinked vertex pairs in $[\gamma]_V$.

A *self-overlapping decomposition* Γ of γ is a vertex pairing such that the induced subcurves are self-overlapping; see Figure 2.4b and Figure 2.4d for examples. The subcurves that result from a vertex pairing are not necessary self-overlapping; see Figure 2.4c. For a self-overlapping decomposition Γ of γ , denote the set of induced subcurves by $\{\gamma_i\}_{i=1}^\ell$. Since each γ_i is self-overlapping, the minimum homotopy area of a self-overlapping decomposition is equal to its sum of the winding area of each subcurve. We define the *area of self-overlapping*

¹also known as the unsigned Gauss code [18, 42]

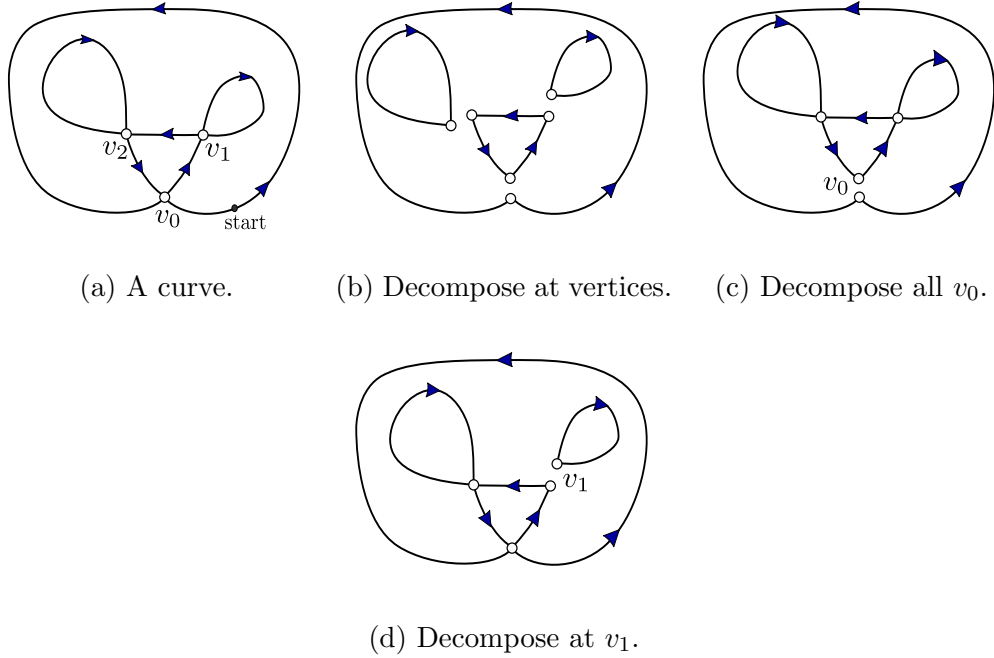


Figure 2.4: Examples and non-examples of self-overlapping decompositions. (a) Curve γ with intersection sequence $\gamma_V = [v_0, v_1, v_1, v_2, v_2, v_0]$. (b) All vertices are paired. (c) One of the subcurves is not self-overlapping. (d) Both subcurves are self-overlapping.

decomposition to be

$$\text{Area}_\Gamma(\gamma) := \sum_{i=1}^{\ell} \text{Area}_W(\gamma_i) = \sum_{i=1}^{\ell} \text{Area}_H(\gamma_i).$$

Fasy, Karakoç, and Wenk [39, 47] proved the following structural theorem.

Theorem 1 (Self-Overlapping Decomposition [39, Theorem 20]). *Any curve γ has a self-overlapping decomposition whose area is minimum over all null-homotopies of γ .*

CHAPTER THREE

FROM CURVES TO WORDS

In order to work with plane curves, one must choose a representation. An important class of representations for plane curves are the various *combinatorial words*. One example is the *Gauss code* [42]. To construct the Gauss code from a curve, assign each crossing a label; then, traverse the curve and record the sequence of labels encountered, possibly also with the direction of how the second strand passes through the first at each intersection. Determining whether a Gauss code corresponds to an actual plane curve is one of the earliest computational topology questions [35].

A plane curve (and its homotopic equivalents) can also be viewed as a word in the fundamental group $\pi_1(\gamma)$ of γ [6, 56, 59]. If we put a point p_i in each bounded face f_i , the curve γ is generated by the unique generators of each $\mathbb{R}^2 - \{p_i\}$. Nie [56] represents curves as words in the fundamental group to find the minimum area swept out by contracting a curve to a point. If the curve lies in a plane with punctures, one can define the crossing sequence of the curve with respect to a system of arcs, cutting the plane open into a simply-connected region. Blank [6] represents curves using a crossing sequences to determine if a curve is self-overlapping. While Blank constructed the words *geometrically* by drawing arcs and Nie defined the words *algebraically*, the dual view between the system of arcs and fundamental group suggests that the resemblance between Blank and Nie's constructions is not a coincidence.

In this section, we describe the construction by Blank; then, we interpret Blank's construction as a way of choosing the basis for the fundamental group under further restriction [59]. We prove that the Blank word is indeed unique when the restriction is enforced, providing clarification to Blank's original definition. We give a complete description

of Nie’s word construction and prove that Nie’s word and Blank’s word are equivalent.

3.1 Blank’s Word Construction

We now describe Blank’s word construction [6, page 5]. Let γ be a generic closed curve in the plane, pick a point in the unbounded face of γ , call it the *basepoint* p_0 . From each bounded face f_i , pick a *representative point* p_i .

Now connect each p_i to p_0 by a simple path in such a way that no two paths intersect each other. We call the collection of such simple paths a *cable system*, denoted as Π , and each individual path π_i from p_i to p_0 as a *cable*.

Orient each π_i from p_i to p_0 . Now traverse γ from an arbitrary *starting point* of γ and construct a cyclic word by writing down the indices of γ crossing the cables π_i in the order they appear on γ ; each index i has a *positive* sign if we cross π_i from right to left and a *negative* sign if from left to right. We denote negative crossing with an overline \bar{i} . We call the resulting combinatorial word over the faces a *Blank word* of γ with respect to Π , denoted as $[\gamma]_B(\Pi)$. Figure 3.1 provides an example of Blank’s construction.

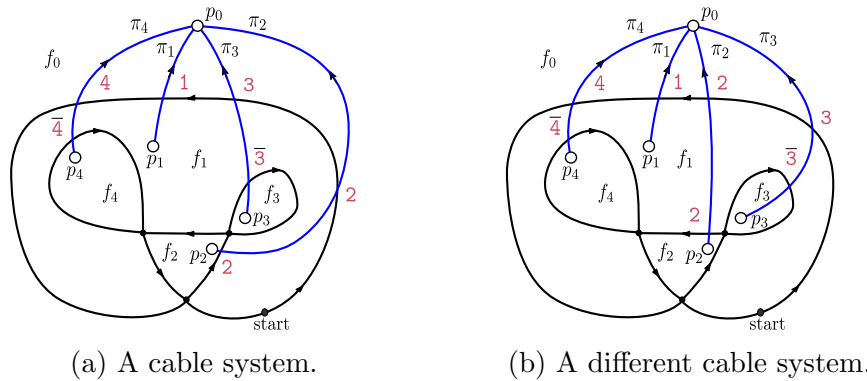


Figure 3.1: An example illustrating the choices made when constructing the Blank word. (a) A curve γ with labeled faces and edges, Π_a is drawn in blue. The Blank word of γ corresponding to Π_a is $[\gamma]_B(\Pi_a) = [2314234]$. (b) The same curve with a different choice of cables Π_b . The corresponding Blank word is $[\gamma]_B(\Pi_b) = [3214324]$.

A word w is *reduced* if there are no two consecutive symbols in w that are identical and with opposite signs. We can enforce every Blank word to be reduced by imposing the following *shortest path assumption*: each cable has a minimum number of intersections with γ among all paths from p_i to p_0 . A simple proof [6, 25] shows that if Π satisfies the shortest path assumption, the corresponding Blank word with respect to Π is reduced. However, the choice of the cable system, and how it affects the constructed Blank word, was never explicitly discussed in the original work (presumably because for the purpose of detecting self-overlapping curves, any cable system satisfying the shortest path assumption works). In general, reduced Blank words constructed from different cable systems for the same curve are not identical, see Figure 3.1a and Figure 3.1b for an example. In this paper, we show that if the two cable systems have the same *cable ordering*—the (cyclic) order of cables around point p_0 in the unbounded face—then their corresponding (reduced) Blank words are the same, under proper assumptions on the cable system.

Our first observation is that the Blank words are invariant under cable isotopy; therefore the cable system can be specified up to isotopy.

Lemma 1 (Isotopy Invariance). *The reduced Blank word is invariant under cable isotopy.*

Proof. Let γ be a curve. Discretize the isotopy of the cables and consider all the possible *homotopy moves* [18] performed on γ and the cables involving up to two strands from γ and a cable, because isotopy disallows the crossing of two cables. No $1 \leftrightarrow 0$ move—the move that creates/destroys a self-loop—is possible as cables do not self-intersect. Any $2 \leftrightarrow 0$ move which creates/destroys a bigon is in between a cable and a strand from γ , which means the two intersections must have opposite signs, and therefore the reduced Blank word does not change. Any $3 \rightarrow 3$ move which moves a strand across another intersection does not change the signs of the intersections, so while the order of strands crossing the cable changes, the order of cables crossed by γ remains the same. Thus the reduced Blank word stays the same. \square

We remark that we can perform an isotopy so that the Blank words are reduced even when the cables are not necessary shortest paths. In the rest of the paper, we sometimes assume Blank words to be reduced based on the context.

Manage the Cable Systems Next, we show that Blank words are well-defined once we fix the choice of basepoint p_0 and the cyclic cable ordering around p_0 , as long as the cables are drawn in a reasonable way. Fix a tree-cotree pair (T, T^*) of γ , where the root of the cotree is on p_0 . We say that a cable system Π is *managed* with respect to the cotree T^* if each path π_i has to be a path on T^* from the root p_0 to the leaf p_i . Given such a collection of cotree paths, one can slightly perturb them to ensure that all paths are simple and disjoint.¹ See Figure 3.2 for examples. Not every cable system can be managed with respect to T^* , see Figure 3.3 for an example.

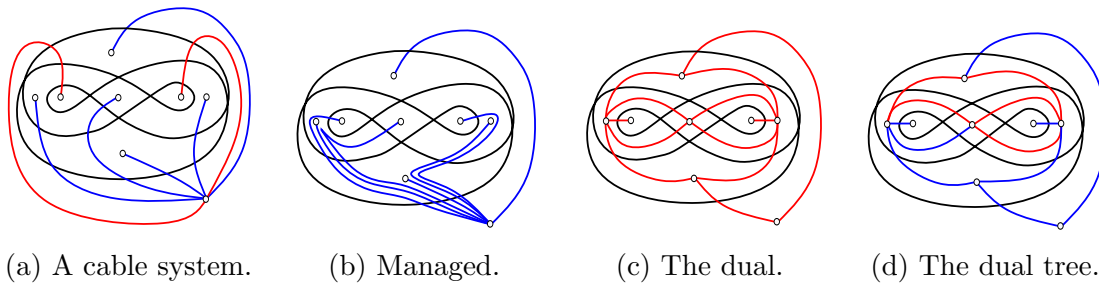


Figure 3.2: An example illustrating how to manage a cable system. (a) A cable system Π_1 on γ that is not managed. The red cables do not follow existing paths to the exterior face. (b) A managed cable system Π_2 on γ . (c) The dual γ^* in red. (d) The spanning tree T^* in γ^* generated by the managed cable system Π_2 .

We now show that if two managed cable systems satisfying shortest path assumption with identical cable ordering around p_0 , their corresponding Blank words are the same. Note that managed cable systems require a fixed tree-cotree pair. We emphasize that the shortest path assumption is necessary; one can construct two (not necessarily shortest) cable

¹In other words, the cables are *weakly-simple* [68].

systems having the same cable ordering but different corresponding reduced Blank words (see Figure 3.4a and Figure 3.4b).

Lemma 2 (Blank Word is Unique). *Given a curve γ , if the basepoint and the cable ordering of a managed cable system that satisfies the shortest path assumption is fixed, then the Blank word of γ is unique.*

Proof. We will argue that once the basepoint p_0 and the order of cables in Π around p_0 is fixed, all the drawings of Π respecting the cable ordering lead to the same Blank word. Because the cable system is managed, the tree-cotree pair of γ are fixed and we can safely contract the primal tree and treat the graph as a collection of nested loops. If the path passes through a vertex of γ , it counts as two crossings. We prove that all the cables to the loops at certain depth have a fixed ordering by induction on the depth. This is sufficient as any two cable systems with the same cables and ordering on every loop of the same depth must be isotopic, thus by Lemma 1 their Blank words are identical. Because of the shortest path assumption, there is only one way to draw the cables to the depth-1 contours.

For any ℓ , imagine all cables of depth at least ℓ are currently drawn from p_0 to the depth- ℓ loops, where on each loop the collection of the cables are precisely those faced contained within the loop, and the cable ordering on the loops is fixed. Due to the shortest path assumption, every cable of depth ℓ has to terminate at their corresponding face. There is at most one unique way to extend each cable of depth ℓ to its representative point in the face while keeping all depth- ℓ cables disjoint and simple, up to isotopy of the cables. (If no such drawing exist, this particular cable ordering is not realizable as a cable system.) By Lemma 1, isotopy does not change the order the curve γ passing through these depth- ℓ cables. Now we partition the cables to faces of depth greater than ℓ based on children loops that contains the corresponding faces. There is one unique way to extend each cable of depth greater than ℓ to the depth- $(\ell + 1)$ loops up to isotopy. Again by Lemma 1, isotopy does not

change the order the curve γ passing through the cables of depth more than ℓ . By induction, the Blank word of γ is unique. □

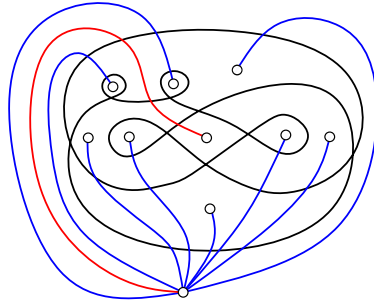


Figure 3.3: An example of a cable system with a cable, in red, that cannot be managed.

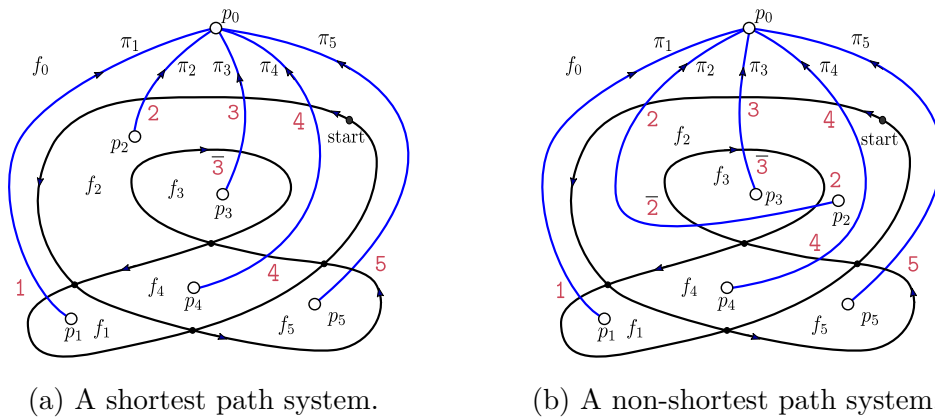


Figure 3.4: (a) A curve with labeled faces and edges. A shortest path cable system Π_a is drawn in blue. From the indicated start position, the Blank Word is $[\gamma]_B(\Pi_a) = [43254\bar{3}1]$. (b) The same curve with a cable system Π_b that does not fulfill the shortest path assumption. The Blank word is $[\gamma]_B(\Pi_b) = [43254\bar{2}321]$.

Therefore, given any plane curve γ , the Blank word is well-defined (if exists), independent of the cable system after specifying a cyclic permutation of all the bounded faces of γ .

3.2 The Nie Word Construction

In an unpublished manuscript [56, 57], Nie described how to compute the minimum homotopy area between any two planar closed curves using the language of geometric group theory. Nie constructed a combinatorial word representing the planar closed curve, followed by performing dynamic programming on the word based on a structure called “foldings” (see Section 4.1). But first, let us describe the word construction.

Choose a point p_i for each bounded face f_i of γ ; denote the collection of points as P . Consider the punctured plane $X := \mathbb{R}^2 \setminus P$ and its fundamental group $\pi_1(X)$. Choose a set of generators Σ for $\pi_1(X)$, where each x_i in Σ represents the generator of $\pi_1(\mathbb{R}^2 \setminus \{p_i\}) \cong \mathbb{Z}$.

Now the fundamental group $\pi_1(X)$ is a free group over such generators, and the curve γ can be represented as a word over generators of $\pi_1(X)$. However, there is more than one way to map each generator of $\pi_1(\mathbb{R}^2 \setminus \{p_i\})$ into $\pi_1(X)$, due to the fact that in order for $\pi_1(X)$ to be a group, one has to choose an endpoint x_0 and turn each closed curve in $\pi_1(X)$ into a closed path connecting to x_0 . Nie never specified the choice of the connecting path because his algebraic formulation always gives the same answer under any mapping of the generators.

Nie’s construction can also be interpreted combinatorially [57]. Again consider the curve γ as a four-regular plane graph. Pick a tree-cotree pair (T, T^*) of γ such that T^* is a BFS-tree; naturally the tree T is contractible. For the sake of illustration, contract T into a single point t ; now each cotree edges is a single closed path at t , enclosing at least one point in P . For our purpose of proving word equivalence, there are two natural sets of generators for $X := \mathbb{R}^2 \setminus P$:

- set of all cotree edges, and
- set of all face boundaries; i.e. sequences of cotree edges around each face containing p_i .

We now describe the change-of-basis between the two sets of generators in graph-theoretic

terms. Traverse γ from some arbitrary starting point and orient each edge of γ accordingly. Now, for each face f_i , define the *boundary operator* ∂ by mapping face f_i to the signed cyclic sequence of edges around face f_i , where each edge is signed positively if it is oriented counter-clockwise and negatively otherwise.

Now, write the curve γ as a cyclic word over the cotree edges T^* by traversing γ , ignoring all tree edges in T . We perform the following procedure inductively on the cotree T^* to construct another cyclic word, this time as an element in the free group over the faces of γ . Starting from the leaves f of T^* , rewrite each edge e bounding the face f (that is, the dual of the unique edge connecting f to its parent in T^*) as a singleton word based on the index of f , with positive sign if edge e is oriented counter-clockwise, or with negative sign otherwise. Next, for any internal node f of T^* , the boundary ∂f consists of a sequence of (1) tree edges, (2) cotree edges to children of f in T^* denoted as e_1, e_2, \dots, e_r , and (3) (a unique) cotree edge to parent of f denoted as e_f :

$$\partial f = [e_f e_1 e_2 \dots e_r].$$

We can now inductively rewrite each child cotree edge e_i as a free word w_i over the faces (and ignore all tree edges). We emphasize that each word for the child cotree edge constructed inductively is a free word, not a cyclic word. Choose a particular but arbitrary way to break the cyclic sequence of faces and rewrite the equation:

$$e_f = \bar{w}_r \cdots \bar{w}_{j+1} \cdot (\bar{w}_j)' \cdot \partial f \cdot (\bar{w}_j)'' \cdot \bar{w}_{j-1} \cdots \bar{w}_1,$$

where $\bar{w}_j = (\bar{w}_j)'(\bar{w}_j)''$ is a particular way of breaking the face word \bar{w}_j into two. This gives us a free word over the faces for edge e_f , and thus by induction we have rewritten γ as a free word over the faces. Finally, we can turn the free word back into a cyclic word, by observing that the cyclic permutation of the constructed free word over the faces does not affect the

element we are getting in $\pi_1(X)$ (but as a side effect of choosing the basepoint p_0 of γ).

We call the resulting signed sequence of faces the *Nie word* and denoted as $[\gamma]_N(\Sigma)$, where Σ is the choices we made when breaking up the cyclic word at each cotree edge, referred to as a *cycle flattening*. Notice that the definition of $[\gamma]_N$ depends on how we choose to break the cyclic edge sequences, and thus is not well-defined without specifying the choices.

3.3 Word Equivalence

Now we are ready to prove that the two words, one defined geometrically and the other algebraically, are in fact equivalent.

Theorem 2 (Word Equivalence). *Let γ be any plane curve. For a Nie word $[\gamma]_N(\Sigma)$ with a fixed cycle flattening Σ , there is a managed cable system Π such that the Blank word $[\gamma]_B(\Pi)$ is equal to $[\gamma]_N(\Sigma)$. Conversely, any managed cable system Π induces a cycle flattening Σ such that $[\gamma]_B(\Pi)$ and $[\gamma]_N(\Sigma)$ are equal.*

Proof. First, fix a tree-cotree pair (T, T^*) for γ such that T^* is a BFS-tree. Orient the edges of the cotree T^* so that it is rooted at some fixed basepoint p_0 . We prove the following statement by induction on the nodes of T^* from leaves to the root, which implies the theorem:

The Blank subword corresponding to any cotree edge e is the same as the Nie subword corresponding to e .

To prove the statement, we will construct the cables in Π gradually from each face to p_0 , at each step stopping at the cotree edge e in T^* . Let f be an arbitrary non-root node in T^* , and edge e be the unique edge from f to its parent in T^* . If f is a leaf, e is the only edge in ∂f that is not in tree T . This means, when we write ∂f using edges not in T , we have $\partial f = \pm e$, with positive sign if e is oriented counter-clockwise and negative sign otherwise. We draw the cable from the representative point in face f to e ; there is only one possible way to draw the cable up to isotopy.

If f is not a leaf, let e_1, \dots, e_r be other non-tree edges on ∂f besides e in counter-clockwise order around ∂f , *flipping their orientation defined by traversing γ if necessary*. By induction hypothesis, the Blank subword of e_i is the same as its Nie subword; denote the Blank (or Nie) subword of e_i as w_i . This suggests that as we traverse e_i , the cables in Π seem is exactly equal to w_i . Now we need to draw the cable π_f from the representative point of f to edge e . By construction of the Nie subword and the given cycle flattening Σ , the Blank subword on e must be of the form

$$\bar{w}_r \cdots \bar{w}_{j+1} \cdot (\bar{w}_j)' \cdot f \cdot (\bar{w}_j)'' \cdot \bar{w}_{j-1} \cdots \bar{w}_1,$$

where $\bar{w}_j = (\bar{w}_j)'(\bar{w}_j)''$ is a particular way of breaking the face word \bar{w}_j into two. (See Figure 3.5.) Because of the shortest path assumption, the collection of symbols inside each w_i corresponds to exactly the faces contained within the region formed by cotree edge e_i and the primal tree T . Thus, we extend all the cables intersecting the edges e_1, \dots, e_r to e , and create the representative point of face f within the subregion of f bounded by the last cable in $\bar{w}_r \cdots \bar{w}_{j+1} \cdot (\bar{w}_j)'$ and the first cable in $(\bar{w}_j)'' \cdot \bar{w}_{j-1} \cdots \bar{w}_1$. This way we can draw the cable π_f so that the corresponding Blank subword is equal to the Nie subword. By induction, we have $[\gamma]_B(\Pi) = [\gamma]_N(\Sigma)$ for the constructed cable system Π , which is managed and satisfies the shortest path assumption by the choice of T^* being a BFS-tree.

From the above construction we can recover a cycle flattening from a given managed cable system satisfying the shortest path property, thus the converse holds as well. \square

Figure 3.6 gives an example demonstrating the one-to-one correspondence, for four different cable systems and cycle flattenings, on the same curve and tree-cotree pair. One consequence coming from the equivalence between two words and Lemma 2 is that Nie word is uniquely determined after knowing the subwords corresponding to cotree edges incident to the unbounded face. This is not obviously from the definition of Nie word itself.

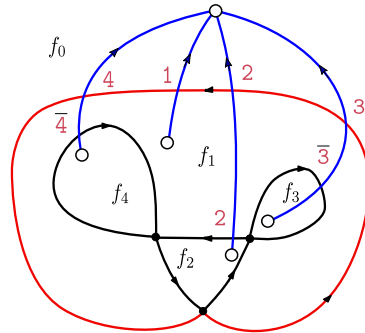


Figure 3.5: As we traverse the red edge γ_r intersects $\mathbf{3}, \mathbf{2}$, then $\mathbf{1}$ —the cable corresponding to f_1 —then $\mathbf{4}$. As we traverse the boundary of f_1 we traverse the red edge, followed by $\overline{\mathbf{4}}$ and $\overline{\mathbf{3}}$. This choice of cable system Π corresponds to cycle flattening at f_1 as $\overline{\mathbf{2332144}}$ by writing $f_1 = \overline{e_2}e_3e_1e_4$, or equivalently, $\overline{\mathbf{2332144}} = \overline{e_2}e_3e_1e_4$. The Blank subword on γ_r , with respect to the cycle flattening, is $e_1 = \overline{\mathbf{32}} \cdot \overline{\mathbf{2332144}} \cdot \overline{\mathbf{4}} = \mathbf{3214}$, as expected.

With this equivalence in hand, for the remainder of the paper we refer to a Nie word or a Blank word of a curve γ as the *word*, denoted as $[\gamma]$ by dropping the subscripts. Keep in mind, however, that the formal equivalence holds only when the cable system is managed.

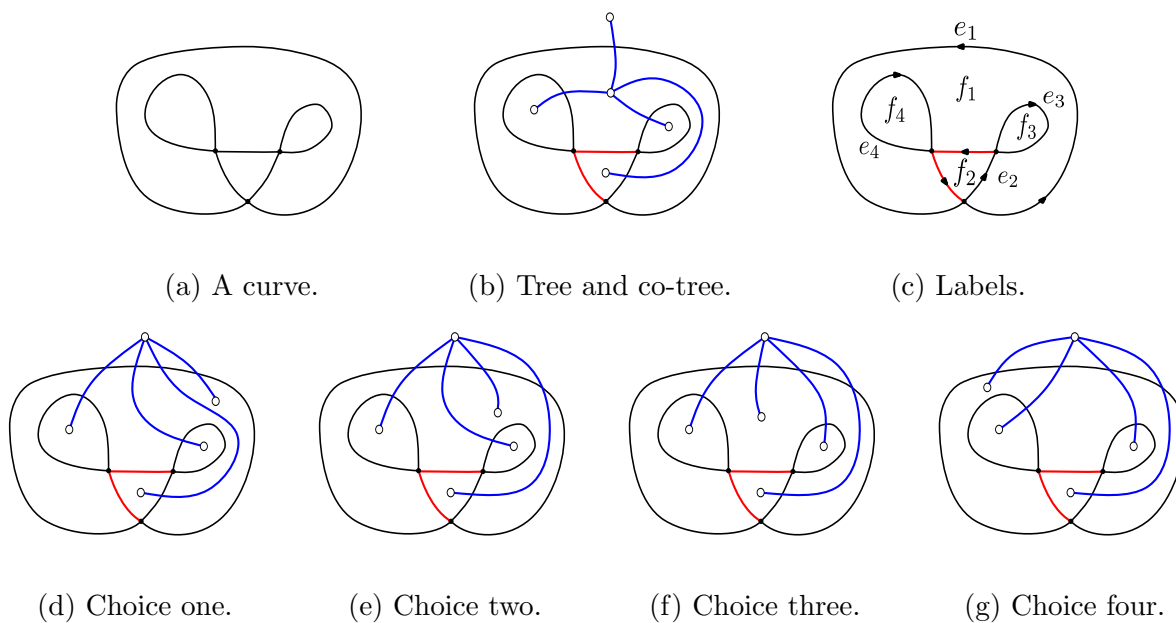


Figure 3.6: An example illustrating the choices made when breaking a cycle. (a) A curve γ . (b) A spanning tree in red and cotree in blue. (c) A labeling of the coedges and faces. We have four ways to break the cyclic face sequence for e_1 , represented using the cable system Π : (d) $e_1 = \partial f_1 e_2 \bar{e}_3 e_4$; (e) $e_1 = e_2 \partial f_1 \bar{e}_3 \bar{e}_4$; (f) $e_1 = e_2 \bar{e}_3 \partial f_1 \bar{e}_4$; and (g) $e_1 = e_2 \bar{e}_3 \bar{e}_4 \partial f_1$.

CHAPTER FOUR

FOLDINGS AND SELF-OVERLAPPING DECOMPOSITIONS

In this section, we give a geometric proof of the correctness to Nie’s dynamic program. To do so, we show that the minimum homotopy area of a curve can be computed from its Blank word using an algebraic quantity of the word called the cancellation norm, which is independent of the drawing of the cables. We then show a minimum-area self-overlapping decomposition can be found in polynomial time.

4.1 The Cancellation Norm and Blank Cuts

Given a (cyclic) word w , a *pairing* is a letter and its inverse $(\mathbf{f}, \bar{\mathbf{f}})$ in w . Two letter pairings, $(\mathbf{f}_1, \bar{\mathbf{f}}_1)$ and $(\mathbf{f}_2, \bar{\mathbf{f}}_2)$, are *linked* in a word if the letter pairs occur in alternating order in the word, $[\cdots \mathbf{f}_1 \cdots \mathbf{f}_2 \cdots \bar{\mathbf{f}}_1 \cdots \bar{\mathbf{f}}_2 \cdots]$. A *folding* of a word is a set of letter pairings such that no two pairings in the set are linked. For example, in the word $[\bar{2}31546\bar{5}4\bar{6}\bar{2}\bar{3}]$ the set $\{(5, \bar{5}), (\bar{3}, 3)\}$ is a folding while $\{(5, \bar{5}), (6, \bar{6})\}$ is not.

The cancellation norm is defined in terms of pairings. The norm also applies in the more general setting where every letter has an associated nonnegative weight. A letter is *unpaired* in a folding if it does not participate in any pairing of the folding. For a word of length m , computing the cancellation norm takes $O(m^3)$ time and $O(m^2)$ space [8, 58]. Recently, a more efficient algorithm for computing the cancellation norm appears in Bringmann *et al.* [10]; this algorithm uses fast matrix multiplications and runs in $O(m^{2.8603})$ time.

The *weighted cancellation norm* of a word w is defined to be the minimum sum of weights of all the unpaired letters in w across all foldings of w [8, 58]. If w is a word where each letter \mathbf{f}_i corresponds to a face f_i of a curve, we define the weight of \mathbf{f}_i to be $\text{Area}(f_i)$. The *area of a folding* is the sum of weights of all the unpaired symbols in a folding. The

weighted cancellation norm becomes $\|w\| := \min_{\mathcal{F}} \sum_i \text{Area}(f_i)$ where \mathcal{F} is the set of all foldings of w and i ranges over all unpaired letter in w .

A dynamic program, similar to the one for matrix chain multiplication, is applied on the word. Let $w = f_1 f_2 \cdots f_\ell$ where $\ell \geq 2$. Assume we have computed the cancellation norm of all subwords with length less than ℓ . Let $w' = f_1 f_2 \cdots f_{\ell-1}$. If f_ℓ is not the inverse of f_i for $1 \leq i \leq \ell - 1$, then f_ℓ is unpaired and $\|w\| = \|w'\| + \text{Area}(f_\ell)$. Otherwise, f_ℓ participates in a folding and there exists at least one k where $1 \leq k \leq \ell - 1$ and $f_k = f_\ell^{-1}$. Let $w_1 = f_1 \cdots f_{k-1}$ and $w_2 = f_{k+1} \cdots f_{\ell-1}$. Then, we find the k that minimizes $\|w_1\| + \|w_2\|$. We have

$$\|w\| = \min\{\|w'\| + \text{Area}(f_\ell), \min_k \{\|w_1\| + \|w_2\|\}\}$$

Nie shows that the weighted cancellation norm whose weights correspond to face areas is equal to the minimum homotopy area using the triangle inequality and geometric group theory. Our proof that follows is more geometric and leads to a natural homotopy that achieves the minimum area.

We now show how to interpret the cancellation norm geometrically. Let $(\mathbf{f}, \bar{\mathbf{f}})$ be a face pairing in a folding of the word $[\gamma]_B(\Pi)$ for some cable system Π . Denote the cable in Π ending at face f as π_f . Cable π_f intersects γ at two points corresponding to the pairing $(\mathbf{f}, \bar{\mathbf{f}})$, which we denote as p and q respectively. Let π'_f be the simple subpath of π_f so that $\pi'_f(0) = q$ and $\pi'_f(1) = p$. We call π'_f a *Blank cut* [6, 37, 52] (see Figure 4.1). Any face pairing defines a Blank cut, and the result of a Blank cut produces two curves each with fewer faces than the original curve: namely, γ_1 which is the restriction of γ from q to p following by the reverse of path π'_f , and γ_2 which is the restriction of γ from p to q followed by path π'_f .

In order to not partially cut any face, we require all Blank cuts to occur along the boundary of the face being cut. When cutting face f_i along path π_j , we reroute all cables crossing the interior of f_i , including π_j but excluding π_i , along the boundary of f_i through an

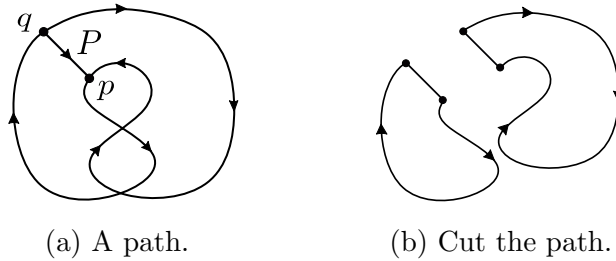


Figure 4.1: An example of a Blank cut. (a) A curve with labeled path P . (b) The two induced subcurves from cutting along P .

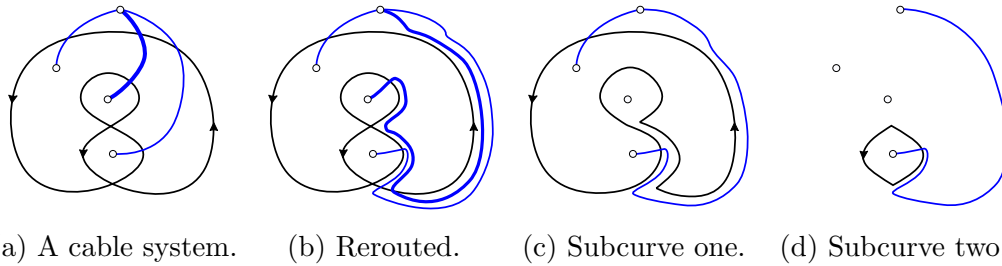


Figure 4.2: An example illustrating how to avoid partially cutting a face. (a) A curve with cables. (b) Isotopy the cables to not partially cut any faces. (c) One subcurve resulting from cutting along the middle cable. The curve is weakly simple and there are two cables in this face. (d) The other subcurve.

isotopy, so that no cables intersect π_i . Lemma 1 ensures that the reduced Blank word remains unchanged. See Figure 4.2 for an example. Notice that different cables crossing f_i might be routed around different sides of f_i in order to avoid intersecting cable π_i and puncture p_i . This way, we ensure the face areas of the subcurves are in one-to-one correspondence with the symbols in the subwords induced by a folding.

Using the concept of Blank cut we can determine if a curve is self-overlapping. A subword σ of w is *positive* if $\sigma = \mathbf{f}_1 \mathbf{f}_2 \dots \mathbf{f}_k$, where each letter \mathbf{f}_i is positive. A pairing $(\mathbf{f}, \bar{\mathbf{f}})$ is *positive* if one of the two subwords of the (cyclic) word w in between the two symbols $\mathbf{f}, \bar{\mathbf{f}}$ is positive; in other words, $w = [\mathbf{f} p \bar{\mathbf{f}} w']$ for some positive word p and some word w' . A folding

of w is called a *positive folding*¹ if all pairings in w are positive, and the word constructed by replacing each positive pairing (including the positive word in-between) $\mathbf{fp\bar{f}}$ in the folding with the empty string is still positive. Words that have positive foldings are called *positively foldable*. Blank established the characterization of self-overlapping curves via Blank cuts.

Theorem 3 (Self-Overlapping Detection [6]). *Curve γ is self-overlapping if and only if γ has rotation number 1 and $[\gamma]_B(\Pi)$ is positively foldable for any shortest Π .*

Blank was also interested in the number of immersions that are not diffeomorphic. In Figure 4.3 we see a curve with two different immersions. The pairings in the groupings are different. Blank showed that the number of different groupings of w is equal to the number of classes of extensions of γ .

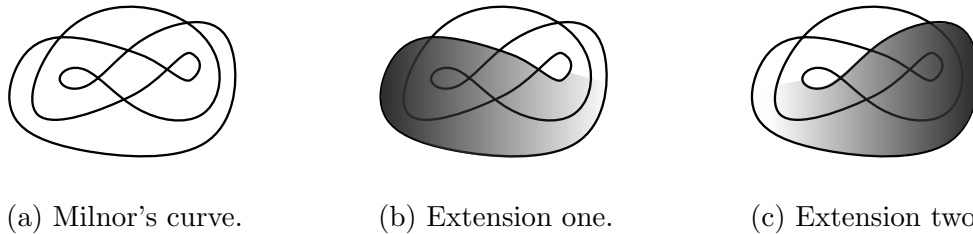


Figure 4.3: An example of a curve with two extensions that are not diffeomorphic. (a) Milnor's curve. A possible word corresponding to this curve is $w = [251346\bar{5}3467253\bar{6}]$. (b) An extension corresponding to the grouping $[251346\bar{5}3467253\bar{6}]$. The blue and green symbols represent pairings. (c) An extension corresponding to the grouping $[251346\bar{5}3467253\bar{6}]$.

We face a difficulty when interpreting Nie's dynamic program geometrically. In our proof we have to work with *subcurves* (and their extensions) of the original curve and the induced cable system. For example, after a Blank cut or a vertex decomposition, there might be multiple cables connecting to the same face creating multiple punctures per face, and cables might not be managed or follow shortest paths to the unbounded face (see Figure 4.2c and Figure 4.7b). In other words, the subword corresponding to a subcurve with respect to

¹Blank called these pairings *groupings*

the induced cable system might not be a regular Blank word (remember that Blank word is only well-defined when the cable system is managed, all cables are shortest paths, and the cable ordering is fixed; see Section 2). To remedy this, we tame the cable system first by rerouting them into another cable system that is managed and satisfies the shortest path assumption, then merging all the cables ending at each face. We show that while such operations change the Blank word of the curve, the cancellation norm of the curve and the positive foldability does not change. We summarize the property needed below.

Lemma 3 (Folding Independence). *Let γ be any curve with two cable systems Π and Π' such that the weights of the cables in Π ending at any fixed face sum up to the ones of Π' . Then, any folding F of $[\gamma](\Pi)$ can be turned into a folding F' of $[\gamma](\Pi')$, such that the area of the two foldings are identical.*

As a corollary, we have the following.

Corollary 1 (Cable Independence). *The minimum area of foldings (the cancellation norm) of $[\gamma](\Pi)$ and the existence of a positive folding of $[\gamma](\Pi)$ are independent of the choice of Π .*

Next, we prove that for each folding there is a homotopy with equal area.

Lemma 4 (Folding to Homotopy). *Let γ be a curve and Π be a managed cable system satisfying the shortest path assumption, and let F be a folding of $[\gamma](\Pi)$. Then, there exists a null-homotopy of γ with area equal to the area of F .*

Proof. We induct on the number of pairings in F . Consider a folding F with k pairings, and let $(\mathbf{f}, \bar{\mathbf{f}})$ be a pairing in F that does not contain any other pairing in between. A Blank cut along this pairing decomposes the curve into two subcurves, at least one of which does not contain any pairings. Call this subcurve γ_1 . We will contract γ_1 to the cut using a homotopy with area equal to the sum of area of all the unpaired letters in the subword of γ_1 ; such homotopy exist by the base case of induction. However, after we contract γ_1 ,

the cable system on the remaining curve might no longer be managed or shortest, and there might be multiple cables in some faces. Therefore we strengthen the inductive hypothesis by assuming that for any subword of $[\gamma](\Pi)$ and its corresponding subcurve γ' of γ with the induced (unmanaged multi-)cable system, and a folding F' with less k pairings on the subword, there exists a null-homotopy of γ' with area equal to the area of the folding F' , and the destination of the null-homotopy can be anywhere on γ' .

Let w' be the subword of $[\gamma](\Pi)$ by removing the subword between the pairing $(\mathbf{f}, \bar{\mathbf{f}})$, folding F' on subword w' be constructed from F by removing the pairing $(\mathbf{f}, \bar{\mathbf{f}})$. Let γ' denote the curve that results from contracting γ_1 to the cut and let Π' denote the cable system induced on γ' by Π . By the (strengthened) induction hypothesis γ' has a null-homotopy with area equal to the one of F' . Combining with the null-homotopy of γ_1 and we are done.

For the base case when F is the empty pairing, if the cables are not managed, construct a managed cable system satisfying the shortest path assumption Π^* for γ and apply Lemma 3; the sum of weights of all letters in the new Blank word remains unchanged. To construct the null-homotopy, decompose the curve into depth cycles by performing a smoothing at each vertex [18], to obtain a null-homotopy with area equal to the depth area. Since each cable in Π^* follows a shortest path to the unbounded face, the number of times an unsigned letter appears in the word is equal to the depth of the face, and thus the homotopy area is equal to the area of the folding. In fact, we can choose any point on the curve to be the destination of the null-homotopy. \square

What we are left with is to prove Lemma 3: Given a curve γ and a cable system Π , if we have a folding on the face word $[\gamma](\Pi)$, there is an equivalent folding of the new word $[\gamma](\Pi')$ that has the same area if we choose a different cable system. As a result, the cancellation norm and the positive foldability of a word are independent to the cable system chosen. Notice that it is sufficient to assume Π' to be a managed cable system with shortest path

assumptions and single cable to each face.

Any two cable systems can always be connected by a sequence of isotopy and order switching between two adjacent cables, followed by a redrawing of the cables induced by a homeomorphism of the plane fixing the punctures $\{p_i\}$, which fixes the cable ordering and the disjointness between cables but the isotopy classes of the cables change. We emphasize that either operation will change the Blank word, but we can always find a folding of the new word that preserves the area.

Next, we show that switching two adjacent cables preserves the area and positivity of the folding.

Lemma 5 (Switch Invariance). *Given curve γ , a cable system, and a folding F , switching the order of two adjacent cables produces another folding on the new word with equivalent area. Furthermore, the new folding is positive if and only if F is.*

Proof. Consider two cables π_f and π_g in a cable system Π that are adjacent in the rotation system. Assume π_g is in the clockwise direction of π_f in the rotation system and we are trying to move π_f across π_g . Let $w := [\gamma]_B(\Pi)$ denote the Blank word before the order of the cables are switched and w' the word after the cables are switched; by Lemma 1 the words are well-defined up to isotopy classes of π_f and π_g . First, we argue that, without loss of generality, we can assume the drawing of π_g follows π_f in an ε -neighborhood before continuing towards puncture p_g , by drawing π_g still counter-clockwise to and ε -close to π_f from p_0 to p_f and back to p_0 , followed by the original drawing of π_g . The new drawing is disjoint from the rest of the cables and is isotopic to the original drawing. This way we can ensure any instance of symbol \mathbf{f} in w is followed immediately by a \mathbf{g} , and any instance of $\bar{\mathbf{f}}$ has a $\bar{\mathbf{g}}$ preceding in the word. (Notice we cannot necessarily say the same about \mathbf{g} because cable π_g continues after reaching p_f .)

Let F be a folding of w , we construct a folding F' of w' with equal area. We separate

into two cases. If no instance of $(\mathbf{f}, \bar{\mathbf{f}})$ is in F , then all pairings in F can be paired in W' and we can set $F' := F$. Now we assume there is an instance of $(\mathbf{f}, \bar{\mathbf{f}})$ in F . We further split into subcases. Look at the two $\mathbf{g}/\bar{\mathbf{g}}$ symbols adjacent to the $(\mathbf{f}, \bar{\mathbf{f}})$ pair in F . If the two \mathbf{g} symbols is a pairing in F , then we can again set $F' := F$. If exactly one of the $\mathbf{g}/\bar{\mathbf{g}}$ adjacent to the $(\mathbf{f}, \bar{\mathbf{f}})$ pair is the paired with another $\bar{\mathbf{g}}$ or \mathbf{g} that is not adjacent to the $(\mathbf{f}, \bar{\mathbf{f}})$ pair in w , then switching the cables naively results in a linked pair,

$$w = [\dots \mathbf{f} \mathbf{g} \dots \overbrace{\mathbf{g} \dots \bar{\mathbf{g}} \bar{\mathbf{f}}} \dots] \longrightarrow w' = [\dots \overbrace{\mathbf{g} \mathbf{f} \dots \mathbf{g} \dots \bar{\mathbf{f}} \bar{\mathbf{g}}} \dots].$$

Instead, after the cables are switched, we pair the \mathbf{g} and $\bar{\mathbf{g}}$ that appear next to the $(\mathbf{f}, \bar{\mathbf{f}})$ pair together in F' :

$$w' = [\dots \overbrace{\mathbf{g} \mathbf{f} \dots \mathbf{g} \dots \bar{\mathbf{f}} \bar{\mathbf{g}}} \dots].$$

The new F' is a folding without linked pairs and does not change the area because the same number and types of symbols are paired. Finally, if both \mathbf{g} and $\bar{\mathbf{g}}$ adjacent to the $(\mathbf{f}, \bar{\mathbf{f}})$ pair are paired with letters not adjacent to $\bar{\mathbf{f}}$ and \mathbf{f} in F , we pair \mathbf{g} next to the \mathbf{f} with the $\bar{\mathbf{g}}$ next to $\bar{\mathbf{f}}$ and the \mathbf{g} not adjacent to the \mathbf{f} with the $\bar{\mathbf{g}}$ not adjacent to $\bar{\mathbf{f}}$ in F' :

$$w = [\dots \overbrace{\mathbf{f} \mathbf{g} \dots \bar{\mathbf{g}} \dots \mathbf{g} \dots \bar{\mathbf{g}} \bar{\mathbf{f}}} \dots] \longrightarrow w' = [\dots \overbrace{\mathbf{g} \mathbf{f} \dots \bar{\mathbf{g}} \dots \mathbf{g} \dots \bar{\mathbf{f}} \bar{\mathbf{g}}} \dots].$$

Again, F' remains a folding without linked pairs and has the same area as F . \square

Next, we show that the cancellation norm does not depend on the cables having the shortest path property. While the face word is unique up to isotopy of the cables based on Lemma 1, two cable systems with identical ordering around p_0 may not be isotopic to each other. The theory of mapping class groups [38, 51] provides tools to convert between all possible isotopy classes of cable systems with same cable ordering.

Let $S_{g,n}$ denote a surface with genus g and n punctures. Let $\text{Diffeo}^+(S_{g,n})$ denote the

group of all orientation-preserving diffeomorphisms from $S_{g,n} \rightarrow S_{g,n}$. Define an equivalence relation \sim on $\text{Diffeo}^+(S_{g,n})$, where $\phi \sim \psi$ if there exists an isotopy between ϕ and ψ . The *mapping class group* of $S_{g,n}$ is the group $\text{MCG}(S_{g,n}) = \text{Diffeo}^+(S_{g,n}) / \sim$. A simple closed curve or a puncture-to-puncture arc λ on a surface is *nonseparating* if the surface remains connected after cutting along λ . A simple closed curve (or arc) in a surface is *essential* if it is not homotopic to a point and not homotopic to a puncture.

Let λ be a simple closed curve in a surface. We now describe a particular mapping class in $\text{MCG}(S_{g,n})$, a *Dehn twist* — about λ . Let A_λ be an arbitrarily thin annulus homeomorphic to the product of \mathbb{S}^1 and the unit interval I ; i.e. $A_\lambda \cong \mathbb{S}^1 \times I$ with $\lambda \cong \mathbb{S}^1 \times \{0\}$. The twist is the map $T_\lambda : A_\lambda \rightarrow A_\lambda$ where $T_\lambda(\theta, t) = (\theta + 2\pi t, t)$ [38]. Intuitively, we can think of T_λ as acting on any path that crosses A_λ by fixing one boundary component of the annulus and rotating the other boundary component one full revolution, dragging all masses within A_λ along. See Figure 4.4 for an example. Dehn twists are well-defined as element in $\text{MCG}(S_{g,n})$ and depend only on the isotopy class of λ [51].

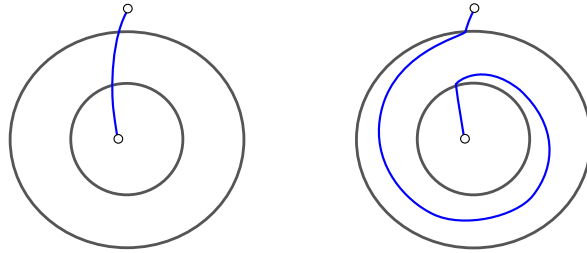


Figure 4.4: An illustration of a Dehn twist.

Let D_n be the closed disk \mathbb{D}^2 with n punctures. The *pure mapping class group* of D_n , $\text{PMCG}(D_n)$, is the subgroup of $\text{MCG}(D_n)$ that fixes all punctures. The group $\text{PMCG}(D_n)$ is generated by a finite number of closed curves.

Theorem 4 (Generators [38, §9.3]). *PMCG(D_n) is generated by Dehn twists about the set*

of simple closed curves that surround exactly two punctures.

Moreover, there are many homotopy classes of curves that surround exactly two points, [38, §9.3] implies any such curve suffices.

Thus, given any cable system we can perform Dehn twists about these generators to obtain another cable system with the same cable ordering such that all cables are shortest paths. We choose a disk in the plane that contains the curve γ , the punctures, and the cables. (One subtlety is that even when the two cable systems have the same *cyclic* ordering, the choice of the disk may produce different *linear* cable ordering when we decide the location to take the containing disk. This can be easily resolved by redrawing some of the cables using isotopy across the infinity.) We now show that performing Dehn twists about suitable simple closed curves does not change the cancellation norm of the word.

Lemma 6 (Twist Invariance). *Let γ be a curve and Π be a cable system, and let F be a folding on the corresponding face word $[\gamma](\Pi)$. Dehn twists about the cables in Π produce another folding on the new word with equal area.*

Proof. Let Π be a cable system on a closed curve γ . Given two punctures p_i and p_j , let $c_{i,j}$ denote the closed curve that contains only p_i and p_j and follows the cables π_i and π_j to the exterior face, then connect. See Figure 4.5a for an illustration. There may be cables between π_i and π_j in the rotation order about p_0 . We bundle all such cables together and denote the corresponding face word B . By Theorem 4, the closed curves $c_{i,j}$ s generate the pure mapping class group. To show that twisting about $c_{i,j}$ does not change the norm, let w be the word generated by the cables before twisting about $c_{i,j}$ and let w' be the word generated by the cables after the twist. We will show that $\|w'\| \leq \|w\|$ and $\|w\| \leq \|w'\|$.

Instances of \mathbf{f}_i and $\bar{\mathbf{f}}_i$ in w become $(\bar{\mathbf{f}}_i \bar{B} \bar{\mathbf{f}}_j B) \mathbf{f}_i (\bar{B} \mathbf{f}_j B \mathbf{f}_i)$ and $(\bar{\mathbf{f}}_i \bar{B} \bar{\mathbf{f}}_j B) \bar{\mathbf{f}}_i (\bar{B} \mathbf{f}_j B \mathbf{f}_i)$ in w' . Instances of \mathbf{f}_j and $\bar{\mathbf{f}}_j$ in w become $(B \bar{\mathbf{f}}_i \bar{B} \bar{\mathbf{f}}_j) \mathbf{f}_j (\mathbf{f}_j B \mathbf{f}_i \bar{B})$ and $(B \bar{\mathbf{f}}_i \bar{B} \bar{\mathbf{f}}_j) \bar{\mathbf{f}}_j (\mathbf{f}_j B \mathbf{f}_i \bar{B})$ in w' . See Figure 4.5 for an example.

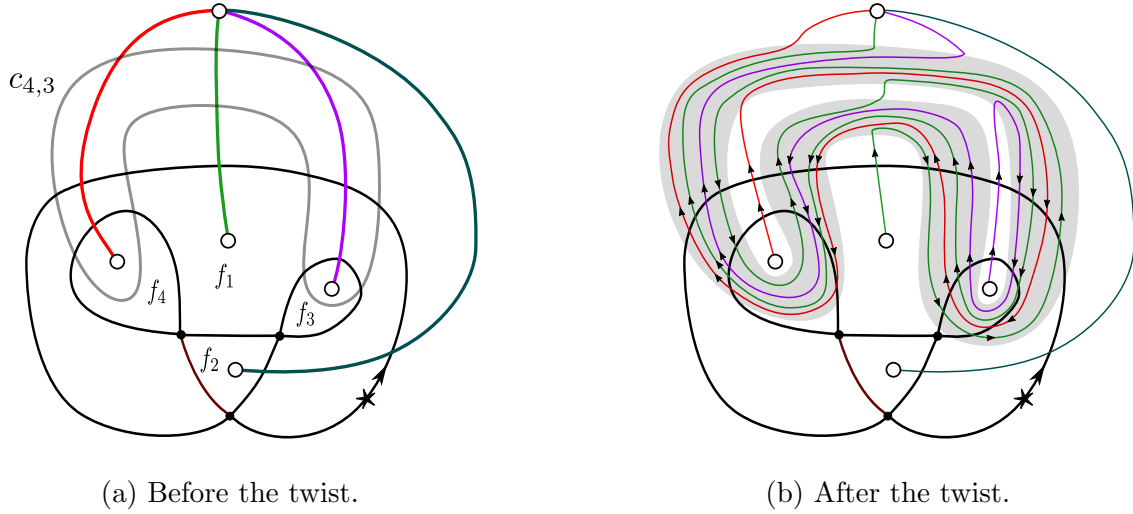


Figure 4.5: (a) An example of the Dehn twist. The curve is shown in black. Starting from the star and following the indicated orientation, the Blank word is $w = [2314234]$. In the instance we have $i = 4$ and $j = 3$; the curve $c_{4,3}$ is shown in gray. The word B is $\mathbf{1}$. (b) After twisting about an annulus formed by fattening $c_{4,3}$ the new word is $w' = [2\overline{14\overline{13}3314\overline{1}1\overline{41314\overline{1}314}2\overline{14\overline{13}3314\overline{1}1\overline{41314\overline{1}314}}]$.

We now construct a folding F' of w' with equal area. All pairs in F are in F' . All letters added by the twist are then paired. If \mathbf{f}_i is in w and unpaired in F then we pair the conjugate letters on each side of \mathbf{f}_i in w' . If \mathbf{f}_i is in w and paired in F then we pair the added letters to the right of \mathbf{f}_i with the added letters to the left of the paired $\overline{\mathbf{f}}_i$ in w' ; similarly, we pair the added letters to the left of \mathbf{f}_i with the added letters to the right of $\overline{\mathbf{f}}_i$. This can be done because the added words are the same mirror pair around symbol \mathbf{f}_i and $\overline{\mathbf{f}}_i$.

$$w' = [\dots (\overline{\mathbf{f}}_i \overline{B} \overline{\mathbf{f}}_j B) \overbrace{\mathbf{f}_i (\overline{B} \mathbf{f}_j B \mathbf{f}_i)} \dots (\overline{\mathbf{f}}_i \overline{B} \overline{\mathbf{f}}_j B) \overline{\mathbf{f}}_i (\overline{B} \mathbf{f}_j B \mathbf{f}_i) \dots]$$

where the overbracket indicates the pairing $(\mathbf{f}_i, \overline{\mathbf{f}}_i)$. Thus, a folding of w' of equal area exist. Because the cancellation norm is computed by taking minimum over all foldings, we have $\|w'\| \leq \|w\|$.

On the other hand, given a folding F' of w' we construct a folding F of w with area at

most the area of F' . If an element of w is unpaired in F' it is also unpaired in F . There are three types of pairings in F' . If both letters in the pair are in w : in which case we add this pair to F . If neither letter in the pair is in w , in which case we do nothing to F . The third type of pairing contains a letter of w paired with a letter in that only presents in w' but not in w ; call this pair $(\mathbf{f}^0, \bar{\mathbf{f}}^1)$ where \mathbf{f}^0 is in w and $\bar{\mathbf{f}}^1$ is in $w' \setminus w$. We will show that there is a unique unpaired letter $\mathbf{f}^k \in w' \setminus w$ that is left unpaired in F , or there is another $\bar{\mathbf{f}}^k \in w$ that we can pair \mathbf{f}^0 with in F .

We construct a sequence of letters in w' that are all \mathbf{f} and $\bar{\mathbf{f}}$ s and terminates with either an unpaired letter in $w' \setminus w$ or a paired letter in w . Let P denote the function that maps a letter to its paired letter in F' , or the identity otherwise. The map C maps a letter of $w' \setminus w$ to its conjugate inverse in the mirror pair added by the Dehn twist if the conjugate inverse has not yet been visited, or the identity otherwise. Note that both P and C are injective, and they only map a symbol \mathbf{f} to its inverse $\bar{\mathbf{f}}$ and vice versa. Beginning with the pairing $(\mathbf{f}^0, \bar{\mathbf{f}}^1)$, we alternate in applying the functions P and C : for any integer i , let $\mathbf{f}^{2i} := C(\mathbf{f}^{2i-1})$ and $\mathbf{f}^{2i+1} := P(\mathbf{f}^{2i})$. Since the word is finite, this unique sequence terminates with either an unpaired letter \mathbf{f}^k of $w' \setminus w$ or an element $\bar{\mathbf{f}}^k$ in w .

- If the sequence of letters ends with an unpaired letter in $w' \setminus w$, this letter uniquely corresponds to \mathbf{f}^0 by following pairings and corresponding inverses in F' . For example, let F' be the folding indicated by the overbrackets, with the underbrackets denoting corresponding conjugate inverse letters,

$$[214133314\bar{1}141314\bar{1}314214133314\bar{1}41314\bar{1}314].$$

The unique unpaired letter corresponding to $\mathbf{3}$ is $\bar{3}$ found by considering the second corresponding inverse letter.

- If the sequence of letters ends with a letter \mathbf{f}^k in w , we pair \mathbf{f}^0 and $\bar{\mathbf{f}}^k$ in F . This does

not create any linked pairs in F since \mathbf{f}^0 and $\bar{\mathbf{f}}^k$ participate in pairings in F' , they are not linked by any other pairing in F' . For example, let F' be the folding indicated by the overbrackets, with the underbrackets denoting corresponding conjugate inverse letters,

$$[\overbrace{214133314\bar{1}141314}^{\quad\quad\quad}\overbrace{1314214133314141314\bar{1}314}^{\quad\quad\quad}].$$

The pairing $(\mathbf{3}, \bar{\mathbf{3}})$ is then added to F .

The sum of the areas of the unpaired letters of F is at most the sum of the areas of the unpaired letters of F' and thus $\|w\| \leq \|w'\|$. This proves the lemma. \square

We show the invariance for positive foldability

Lemma 7. *Being positively foldable does not depend on Dehn twists.*

Proof. Let w be a word and let w' be the word after a Dehn twist about the simple closed curve $c_{i,j}$ as in the proof of Lemma 6. Suppose w is positively foldable with folding F . Then, the folding F' described in Lemma 6 is a positive folding of w' .

On the other hand, suppose w' has a positive folding F' . Then, each element $\bar{\mathbf{f}}$ of w' is paired with an element \mathbf{f} . If both $\bar{\mathbf{f}}$ and \mathbf{f} are in w then we pair them. If neither $\bar{\mathbf{f}}$ nor \mathbf{f} are in w then we ignore them. If one of $\bar{\mathbf{f}}$ and \mathbf{f} are in w and the other is added to w' by the twist, then, since letters are added to w' in pairs, there is another element of w paired with an element of $w' \setminus w$. We can pair the two element in w and ignore the two in $w' \setminus w$ because the pairings in F' are not linked. The resulting pairings give a positive folding of w . \square

Together, Lemma 5 and Lemma 6 imply the cancellation norm does not depend on the isotopy class of the cables. Moreover, Lemma 5 and Lemma 7 imply the positive foldability of the word does not depend on the isotopy class of the cables. One last subtlety: we show that we can handle multiple cables per face.

Handling multiple cables per face. When a face f in a subcurve contains multiple cables and punctures, we treat the punctures as distinct when applying Lemma 5 and Lemma 6. Now, once all the cables follow the cotree shortest paths, consider all cables ending in an arbitrary face f . If they are separated by other cables in the cyclic ordering, we use Lemma 5 and isotopy to move all the cables not ending in f out of the way until they no longer intersect the face. This makes sure that all the cables ending in f are gathered together in the cyclic ordering. Because they follow the same unique cotree path to the exterior, we can merge them into a single cable, where the corresponding symbols in the word are merged into a single symbol with the combined weight. Thus, the subwords induced by a face pairing correspond to the subcurves generated by the Blank cut.

4.2 Compute Min-Area Homotopy from Self-Overlapping Decomposition

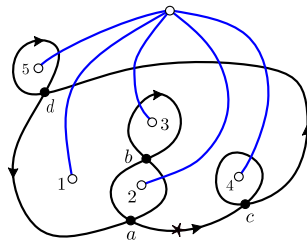


Figure 4.6: A curve with combined word $[c4c4231d5da2b3ba]$.

A self-overlapping decomposition is a vertex decomposition where each subcurve is self-overlapping [39]. By Theorem 1, there exists a self-overlapping decomposition and an associated homotopy whose area is equal to the minimum homotopy area of the original curve.

In order to relate vertex decompositions and face decompositions, we define a word that includes both the faces and vertices. Given any curve γ and cable system Π , traverse γ and record both self-crossings and (signed) cable intersections; we call the resulting sequence of vertices and faces the *combined word* $[[\gamma]](\Pi)$. See Figure 4.6 for an example.

We now show that every self-overlapping decomposition (with respect to the vertex word of γ) determines a folding (of the face word of γ) using the combined word.

Theorem 5 (Self-Overlapping Decomposition to Folding). *Given a self-overlapping decomposition Γ and a cable system Π of γ , there exists a folding F of $[\gamma](\Pi)$ whose area is $\text{Area}_\Gamma(\gamma)$.*

Proof. Begin with the combined word $[[\gamma]](\Pi)$. Decompose $[[\gamma]](\Pi)$ at the vertices given by the self-overlapping decomposition. Let $\Gamma = \{\gamma_1, \gamma_2, \dots, \gamma_s\}$ be the self-overlapping subcurves and $[[\gamma]](\Pi)_i$ be the corresponding subwords of $[[\gamma]](\Pi)$. If we remove the vertex symbols and turn each $[[\gamma]](\Pi)_i$ into a face word $[\gamma_i]'$, such word may not correspond to Blank words of the subcurves; indeed, when decomposing γ into subcurves by Γ , the subcurve along with the relevant cables may contain multiple cables per face and cables might not be managed or follow shortest paths. See Figure 4.7 for an example. However, we can first tame the cable system by choosing a new managed cable system Π^* where the cables follow shortest paths and has one cable per face (as in Section 3.1). Lemma 3 ensures that the cancellation norm and positive foldability of the subcurve remain unchanged. Denote the new face word of γ_i with respect to Π^* as $[\gamma_i] = [\gamma_i](\Pi^*)$.

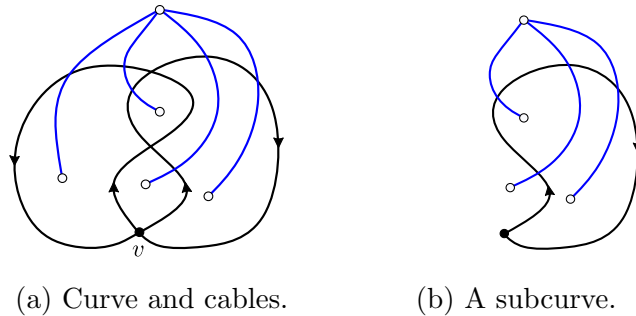


Figure 4.7: We decompose the curve in (a) at vertex v into self-overlapping subcurves, the cable system on the induced subcurve in (b) has more than one marked point in a face and cables do not follow shortest paths.

Since each γ_i is a self-overlapping subcurve in Γ , we can find a positive folding F_i of $[\gamma_i]$

by Theorem 3, and the minimum homotopy area of γ_i is equal to the area of folding F_i . Now Lemma 3 implies that the subword $[\gamma_i]'$ from the original combined word also has a positive folding F'_i whose area is equal to the minimum homotopy area of γ_i . By combining all foldings F'_i of each face subword $[\gamma_i]'$, we create a folding F for $[\gamma](\Pi)$ (no pairings between different F'_i s can be linked). The area of folding F is equal to the sum of areas of foldings F'_i , which in turns is equal to $\sum_i \text{Area}_H(\gamma_i)$, that is, the homotopy area of self-overlapping decomposition $\text{Area}_\Gamma(\gamma)$. This proves the theorem. \square

Corollary 2 (Geometric Correctness). *The dynamic programming algorithm computes the minimum-area homotopy for any curve.*

Proof. By Theorem 1, there exists a self-overlapping decomposition with minimum homotopy area. By Theorem 5, some folding achieves a minimum area. Using Lemma 4, the minimum-area folding produces a minimum-area homotopy. \square

4.3 Min-Area Self-Overlapping Decomposition in Polynomial Time

Finally, we show how to construct a self-overlapping decomposition from a maximal folding with equal area. Before we begin the proof, we include definitions that will help describe the types of curves we encounter. A curve γ is a *k -stack* if it has rotation number k , all bounded faces have positive winding numbers, and $\text{Area}_H(\gamma) = \text{Area}_W(\gamma)$ (see Figure 4.8a for any example). Any k -stack has a vertex decomposition into k self-overlapping curves [36, Theorem 2.15].² A curve is a *$-k$ -stack* if its reversal is a $+k$ -stack. We called a curve γ a *stack* if γ is a $\pm k$ -stack. A curve is *good* [36] if the depth of each face is equal to the absolute value of the winding number; any good curve must have $\text{Area}_D(\gamma) = \text{Area}_H(\gamma) = \text{Area}_W(\gamma)$ by Equation 2.1. Good curves are almost stacks, except that some faces might not have

² k -stacks are called *interior-boundaries* by Titus [67] and the terminology was used in various previous work [36, 39, 52]. The name *k -stack* is well-justified as such curve is the boundary of a stack of k disks [36].

positive winding numbers. A vertex of γ is *sign-changing* if the four incident faces have winding numbers $[1, 0, -1, 0]$ in cyclic order (see Figure 4.8b for an example). If we smooth a sign-changing vertex of a good curve, the two induced subcurves remain good. We can always decompose a good curve into a collection of stacks at all the sign-changing vertices [36, Theorem 5.7].

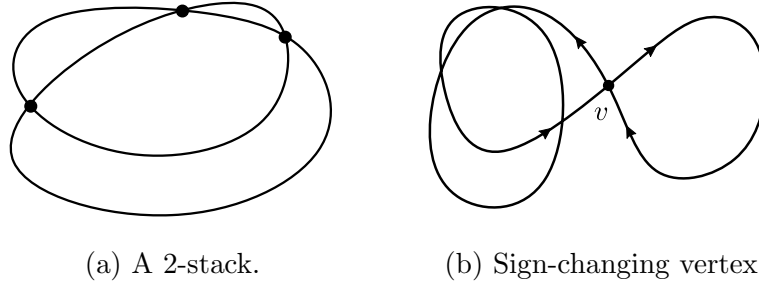


Figure 4.8: (a) A k -stack where $k = 2$. Smoothing at any of the three vertices gives a self-overlapping decomposition. (b) The vertex v is a sign-changing vertex.

With these definitions in hand, we now show how to construct a self-overlapping decomposition from a maximal folding with equal area.

Theorem 6 (Folding to Self-Overlapping Decomposition). *Let γ be a curve and Π be a cable system. Given a maximal folding F of $[\gamma](\Pi)$, there is a self-overlapping decomposition of γ whose area is equal the area induced by the folding F .*

Proof. Let F be any maximal folding of $[\gamma](\Pi)$. Without loss of generality we can assume that the cable system does not cut through the interior of any face by rerouting the cables similar to Section 4.1. By Lemma 1 the Blank word remains unchanged.

Let γ_g be an arbitrary subcurve generated after Blank cutting along the pairings in F . The curve γ_g is must be good: otherwise, there is a face with winding number not equal to its depth, and thus γ_g would cross the corresponding cable from left to right and from right to left. Therefore, we can introduce an extra pair into the folding and F remains unlinked; thus F would not be maximal. Decomposing at all the sign-changing vertices [36,

Theorem 5.7] turns γ_g into a collection of stacks, each of which can be further decomposed into collection of self-overlapping curves [36, Theorem 2.15].

The area of the folding F is equal to the number of unpaired faces in $[\gamma](\Pi)$, which is also equal to the sum of depth area of each good subcurve. Since each subcurve γ_g of folding is good, the depth area of γ_g is equal to its winding area. Any additional decomposition of γ_g into self-overlapping curves respects the additivity of winding areas. Thus, the area of folding F is equal to the area induced by our chosen self-overlapping decomposition. \square

The above theorem implies a polynomial-time algorithm to compute a self-overlapping decomposition with minimum area. Namely, apply the dynamic programming algorithm to compute the minimum-area folding F for $[\gamma](\Pi)$ with respect to some cable system Π . By Theorem 5 the area of F is equal to the minimum homotopy area of γ , and so does the corresponding self-overlapping decomposition given by Theorem 6. Thus, we have the following.

Corollary 3 (Polynomial Optimal Self-Overlapping Decomposition). *Let γ be a curve. A self-overlapping decomposition of γ with area equal to minimum homotopy area of γ can be found in polynomial time.*

CHAPTER FIVE

AN ALGORITHM

In this chapter, we describe the first polynomial time implementation of an algorithm to compute the minimum homotopy area of a piecewise linear closed curve in the plane. We discuss how to use minimum homotopy area as a measure of curve similarity. We then compare minimum homotopy area to the popular measure of curve similarity, the Fréchet distance. We include experiments that compare the runtime of our homotopy area algorithm to an implementation of the Fréchet distance.

5.1 A Polynomial Implementation

Our minimum homotopy area algorithm uses the word equivalence described in Section 3 to construct the Blank word and then compute the cancellation norm. The implementation is written in C++ and uses the Computational Geometry Algorithms Library (CGAL) [40]. The curve is stored as an arrangement of segments in a doubly connected edge list (DCEL) data structure [22]. Given a curve consisting of n segments and $|F|$ bounded faces, our algorithm runs in $O(n^2 + |F|^6)$.

A curve is a four-regular planar graph; Euler's formula implies the number of edges is twice the number of vertices and the number vertices is one less than the number of faces (not counting the unbounded face). Let w be a Blank word, the number of times a face symbol appears in w is equal to the depth of the face. The depth of each face is $O(|F|)$ and thus, the length of the word is $O(|F|^2)$. See Figure 5.1 for an example of a curve with a word of length $\Omega(|F|^2)$. The cancellation norm is cubic in the length of the word, so once we construct the word, the runtime is $O(n^2 + |F|^6)$. Recently, in [10], Bringmann et al. showed that the cancellation norm can be computed in $O(|F|^{2.8603})$ giving a total theoretical runtime

of $O(n^2 + |F|^{5.721})$ but our current implementation does not include this improvement. It is an open questions whether this runtime can be improved.

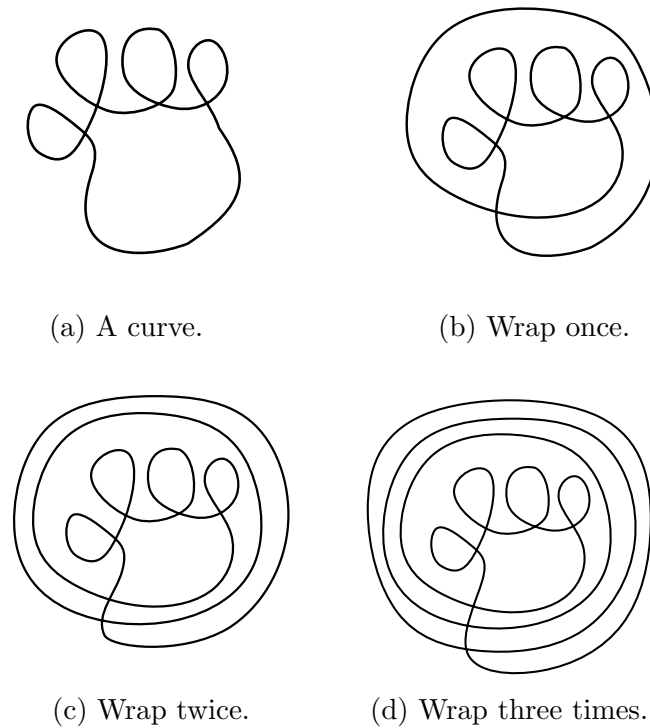


Figure 5.1: An example of a curve with a Blank word $\Omega(|F|^2)$. (a) A curve with Blank word $[12345]$. (b) Each time we wrap the curve, each of the five faces appears in the word again. The word is now $[12345123456]$. (c) Wrapping again makes the word $[123451234561234567]$. (d) Wrapping one more time makes the word $[12345123456123456712345678]$. There are $\frac{|F|}{2}$ faces each with depth $\frac{|F|}{2}$ and the length of the word is $\Omega(|F|^2)$.

Pseudocode of the implementation is given in Algorithm 5.1. We construct two non-trivial data structures, one to contain information about the faces and one to contain information about the edges. For each face, we record its discover time in a BFS search of the dual from the unbounded face, area, an edge between our current face and an adjacent face with lesser depth and a flag to determine if we have laid the cable for this face. For each edge, we record a vector of cables passing through that edge (if any) and an orientation.

Algorithm 5.1: PolynomialMinimumAreaHomotopy(γ)

Input: A list of points in \mathbb{R}^2 with the first point equal to the last point

Output: The minimum homotopy area

- 1: Build an arrangement of segments
- 2: Breadth first search on the dual and label each face with its discover time
- 3: Compute the area of each face
- 4: Use the depth to label a edge toward the exterior for each face
- 5: Traverse the curve and orient each edge
- 6: **for all** Each face starting with the deepest **do**
- 7: **while** Not at the exterior face **do**
- 8: **if** The face has not been visited **then**
- 9: Mark the face as visited
- 10: Add a cable for this face to the current vector of cables
- 11: **end if**
- 12: Add the current vector of cables to the edge corresponding to the current face
- 13: **end while**
- 14: **end for**
- 15: Traverse the curve and read off the cables that intersect each edge
- 16: Compute the cancellation norm
- 17: **return** The minimum homotopy area

Now, to formally justify the runtime. In Line 1, we build the arrangement of segments. We use the fact that we are inserting x -monotone segments and the run-time of insertion of a segment is proportional to the complexity of the zone of each segment [66]. The zone of a segment is $\Theta(n^2)$ in the worst case. In Line 3, we compute the area of each face. To do this, we traverse the boundary of each face and use the formula described in [7]. Each edge is on the boundary of at most two faces so the runtime is $\Theta(n)$. We visit each edge at most twice to traverse the boundary of each face. The faces are weakly simple, but the formula still applies. For each face, we identify an edge on its boundary that is adjacent to a face with lesser depth. We do this in Line 4 and takes $\Theta(n)$ time. Orienting the curve in Line 5 takes $\Theta(n)$ time. Laying the cables in Line 6 takes $\Theta(|F|^2)$ time, for each face with visit at most its depth number of faces. In Line 15, we traverse the curve again and record the word in $\Theta(n + |F|^2)$ time. Finally, we compute the cancellation norm in Line 16 to obtain a total runtime of $O(n^2 + |F|^6)$.

There are two bottlenecks in the run time. First, building the arrangement takes $O(n^2)$ in the worst case, but if the segments are short the zone of each segment is unlikely to be linear in complexity. Second, computing the norm of a long word is expensive. Often, curves do not have $O(n)$ faces each with depth $O(n)$ as in Figure 5.1.

5.2 Curve Similarity

Comparison of shapes is essential in computer vision, robotics, computer aided design, and medical imaging. Curves and paths are a particularly important class of shapes. In Geographic Information Systems (GIS), curves are often presented as a sequence of linearly interpolated measurement points. In [16], Chambers and Wang present minimum homotopy area as a natural measure of curve similarity. They construct a closed curve from two paths as follows. Let P and Q be two piecewise linear paths in \mathbb{R}^2 . Let $\text{rev}(Q)$ denote the reversal of Q . If P and Q have the same end points, we construct a closed curve by

defining $\gamma = P \circ \text{rev}(Q)$. The distance between P and Q is the minimum homotopy area of γ . If P and Q do not have the same endpoints, we add two additional segments to construct a closed curve by connecting the initial and terminal points. Let n be the number of points on P and let m be the number of points on Q . Denote the initial points on P and Q by P_0 and Q_0 be respectively and let P_n and Q_m denote the end points. Let $\sigma_s = \overline{Q_0 P_0}$ and let $\sigma_t = \overline{P_n Q_m}$. The let $\gamma = P \circ \sigma_t \circ \text{rev}(Q) \circ \sigma_s$ and the distance P and Q is the minimum homotopy area of γ .

One of the first measures of curve similarity is the Hausdorff distance [3]. The Hausdorff distance identifies P and Q with the set of points lying on these curves. Let $\|\cdot\|$ denote the L_2 norm. The *directed Hausdorff distance* from P to Q is

$$\vec{d}_H(P, Q) := \max_{x \in P} \min_{y \in Q} \|x - y\|,$$

and the *Hausdorff distance* between P and Q is

$$d_H(P, Q) := \max \left(\vec{d}_H(P, Q), \vec{d}_H(Q, P) \right).$$

The Hausdorff distance between two curves can be computed in $O((n + m) \log(n + m))$ [3]. Intuitively, two sets have a small Hausdorff distance if every point of both sets is close to some point in the other set. The Hausdorff distance considers the curves as sets of points and is insensitive to the ordering along the curves. There are curves that are quite different that have small Hausdorff distance. An example is given in Figure 5.4d. For this reason, the Hausdorff distance is no longer commonly used as a measure of curve similarity. A measure that respects the structure of the curves is the Fréchet distance.

5.3 Fréchet Distance

The most popular measure of curve similarity in the computational geometry literature is the Fréchet distance. The *Fréchet distance* between two paths P and Q is:

$$d_F(P, Q) := \inf_{\alpha, \beta: [0,1] \rightarrow [0,1]} \max_{t \in [0,1]} \|P(\alpha(t)) - Q(\beta(t))\|, \quad (5.1)$$

where α and β range over all continuous monotonically increasing functions [2]. Intuitively, imagine a person traversing one path and their dog traversing the other path on a leash. The speed of both the dog and the person can vary but neither can backtrack. The Fréchet distance between the paths is the length of the shortest leash needed for both the person and the dog to traverse their respective curves from start to finish without backtracking. See Figure 5.2 for an illustration. A detailed comparison between the Hausdorff distance and the Fréchet distance is given in [5].

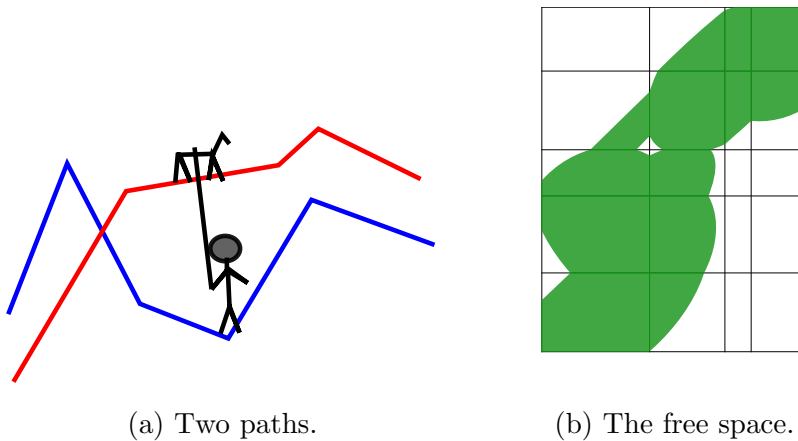


Figure 5.2: An illustration of two paths and their corresponding free space diagram. (a) A person and their dog realizing the Fréchet distance between two paths. (b) The free space between the two paths. The free space diagram was generated using the applet of Schäfer [63].

Alt and Godau introduced the Fréchet distance and gave an algorithm to compute it in $O(nm \log(nm))$ time and $O(nm)$ space [2]. Their algorithm is the basis for many

extensions and variations of the Fréchet distance. We give a brief describe their algorithm. Consider the simpler case where P and Q each consist of a single segment. A point for the human and a point for the dog can be represented as a point in the unit square $[0, 1] \times [0, 1]$. We define a point $(s, t) \in [0, 1] \times [0, 1]$ to be *free* if $d_F(P(s), Q(t)) \leq \epsilon$. Then, $d_F(P, Q) \leq \epsilon$ if there is a free monotone path from $(0, 0)$ to $(1, 1)$. Alt and Godau made the key observation that the free space for two segments is convex, namely the intersection of the unit square with an ellipse. See Figure 5.3 for an illustration. Thus, one only needs to check the distance between the end points of the segments, if they are both within a distance of ϵ then the Fréchet distance between the segments is less than ϵ .

Now to apply this idea to curves that consists of more than one segment. For each pair of segments, one has a cell in a grid and an free ellipse. Moreover, neighboring cells have free ellipsis that meet consistently on the boundary of the cells. This collection of grid cells and free ellipsis is the *free space diagram*. See Figure 5.2b for an example. One then determines if there is a monotone path from $(0, 0)$ to $(1, 1)$ to determine if $d_F(P, Q) \leq \epsilon$. Dynamic programming is used to compute the free space diagram and determine if there is a free monotone path to answer the decision version of the Fréchet distance in $O(mn)$ time. To determine the Fréchet distance between two curves one can use parametric search to compute the distance in $O(nm \log(nm))$ time.

5.4 Experiments

In this section, we compare the experimental runtime of our minimum homotopy area algorithm to an implementation of the Fréchet distance. The implementation of the Fréchet distance is in python and can be found <https://pypi.org/project/frechetdist/> [32]. Our minimum homotopy area code can be found https://github.com/Brad-A-M/minimum_homotopy_area. Our experiments show that our minimum homotopy area algorithm is faster in the cases we considered.

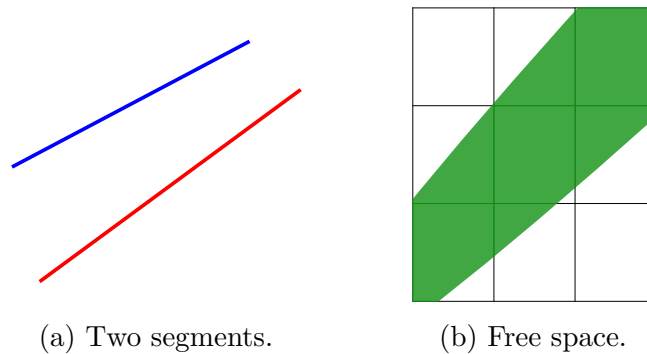


Figure 5.3: An example showing the free space of two paths. (a) Two paths that each consist of a single segment. (b) The free space between the two segments is convex. The free space diagram was generated using the applet of Schäfer [63].

The Fréchet distance computes a distance between two paths and homotopy area requires a closed curve. In order to use minimum homotopy area as a measure of curve similarity we connect the end points of two paths as described in Section 5.2. We used standard hardware for our experiments. Specifically, a mac laptop with 1.6 GHz Dual-Core Intel Core i5 processor. The time to compute the similarity of the curves in Figure 5.4 is given in Table 5.2. For the curve given in Figure 5.4a, homotopy area is 83 times faster. Homotopy area was even faster for the curve shown in Figure 5.4d that has 81 faces. We emphasize that our algorithm is implemented in C++ while the Fréchet distance is in Python and this contributes to the difference in runtime. Our experiments suggest that it is reasonable to use minimum homotopy area as a measure of curve similarity.

5.5 Discussion

We now discuss differences between minimum homotopy area and the Fréchet distance as a measure of curve similarity. The Fréchet distance has the advantage of being well established. Techniques have been developed to compute the Fréchet distance quickly in practice, for instance, in [11], Bringmann et al. describe how to avoid building the entire

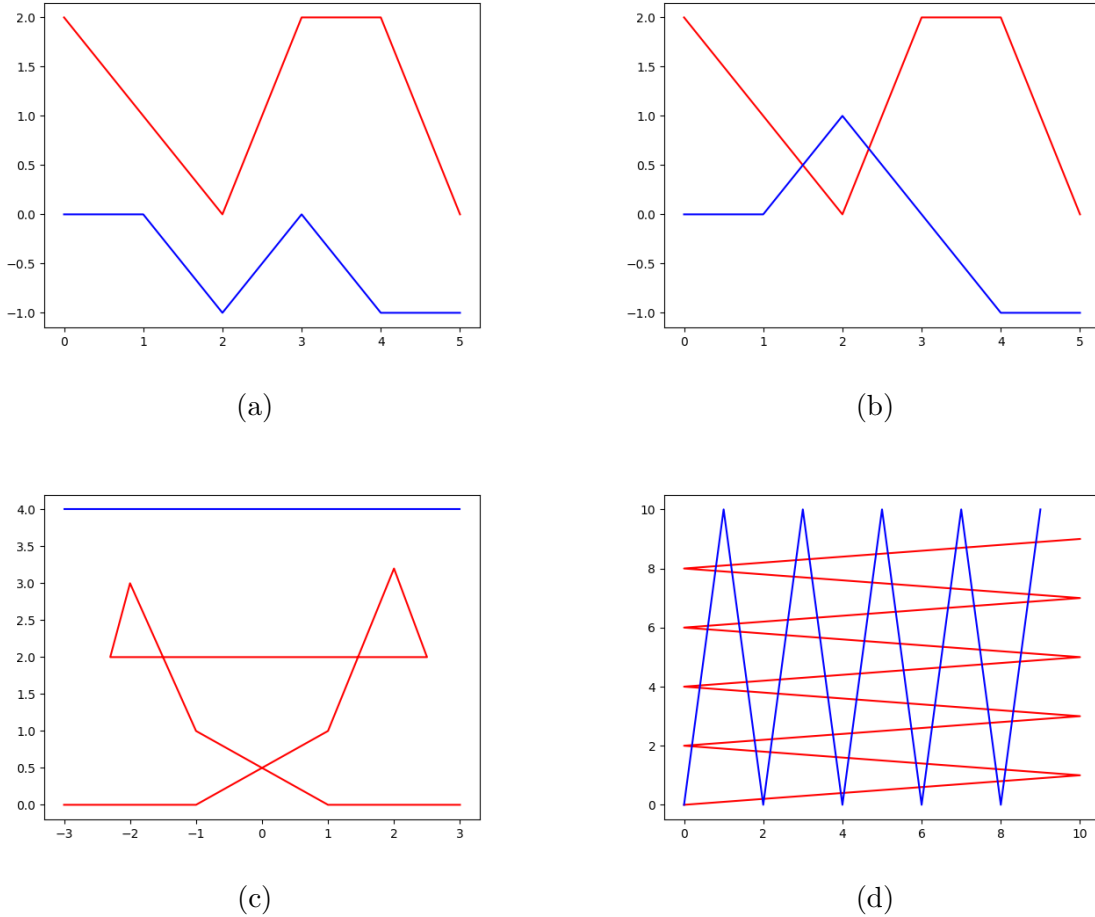


Figure 5.4: Four pairs of paths are shown. The homotopy area is of the closed curve formed by connecting the endpoints of the paths. (a) Paths P and Q with Fréchet distance 2.235 and minimum homotopy area 8.5. (b) Paths P and Q with Fréchet distance is 2.235 and minimum homotopy area is 7.333. (c) Paths P and Q with Fréchet distance 4.123 and minimum homotopy area is 4.00455. (d) Paths P and Q with Fréchet distance 10.050 and minimum homotopy area is 72.0663.

free space diagram in most cases. The challenge of the ACM SIGSPATIAL GIS Cup 2017 was to implement a near-neighbor data structure under the Fréchet distance. The Fréchet distance has been generalized in a number of ways. For example, the weak Fréchet distance allows the dog and the owner to backtrack in order to possibly use a shorter leash [2].

Efforts to compute the Fréchet distance in subquadratic time have been explored [1, 9].

Table 5.1: The coordinates for the paths shown in Figure 5.4.

Figure	Paths
Figure 5.4a	$P = [(0, 2), (1, 1), (2, 0), (3, 2), (4, 2), (5, 0)]$ $Q = [(0, 0), (1, 0), (2, -1), (3, 0), (4, -1), (5, -1)]$
Figure 5.4b	$P = [(0, 2), (1, 1), (2, 0), (3, 2), (4, 2), (5, 0)]$ $Q = [(0, 0), (1, 0), (2, 1), (3, 0), (4, -1), (5, -1)]$
Figure 5.4c	$P = [(-3, 4), (-2, 4), (-1.5, 4), (-1, 4), (0, 4), (0.5, 4), (1, 4), (1.5, 4), (2, 4), (3, 4)]$ $Q = [(-3, 0), (-1, 0), (1, 1), (2, 3.2), (2.5, 2), (-2.3, 2), (-2, 3), (-1, 1), (1, 0), (3, 0)]$
Figure 5.4d	$P = [(0, 0), (10, 1), (0, 2), (10, 3), (0, 4), (10, 5), (0, 6), (10, 7), (0, 8), (10, 9)]$ $Q = [(0, 0), (1, 10), (2, 0), (3, 10), (4, 0), (5, 10), (6, 0), (7, 10), (8, 0), (9, 10)]$

A lower bound of $\Omega(n \log n)$ for computing the Fréchet distance is given in [12]. Extending the Fréchet distance to compute a distance between surfaces has been considered in [13] and [14]. The Fréchet distance has been applied to problems outside of geometry, for instance, folded 3D structures of biomolecules [46]. Computing the minimum Fréchet distance between two curves under translation is explored in [4] and, allowing translation and rotation, is explored in [70]. Extending minimum homotopy area to allow translation, rotation, and scaling is an interesting direction of future work.

A significant limitation of the Fréchet distance is that it is sensitive to outliers. Local noise frequently occurs in real data. For example, in Figure 5.5, a local error occurred when the author was using a popular application to track a run with GPS. A single error in a GPS point produces a spike in the curve. Thus, the change in the minimum homotopy area is often much smaller than the change in the Fréchet distance. One solution to dealing with local noise is to remove outliers on the curve with edit operations [30, 41].

Minimum homotopy area is able to detect subtle differences in curves that the

Table 5.2: The time, in seconds, to compute the Fréchet distance and the minimum homotopy area for each pair of paths given above.

Paths	Fréchet	Homotopy
Figure 5.4a	0.13696 s	0.00167 s
Figure 5.4b	0.14765 s	0.00259 s
Figure 5.4c	0.13433 s	0.00463 s
Figure 5.4d	0.17228 s	0.03358 s

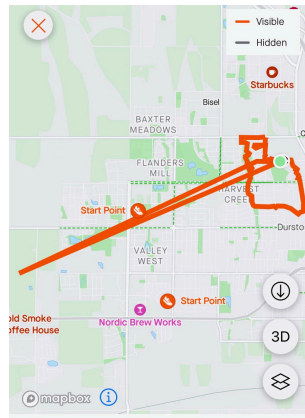


Figure 5.5: A trajectory illustrating an error that significantly changes the Fréchet distance but only slightly changes the homotopy area. Permission to include image granted by [44].

Fréchet distance is unable to distinguish. An example can be seen in Figure 5.4a and Figure 5.4b. These two pairs of curves have the same Fréchet distance but the minimum homotopy area is different.

A major advantage of the Fréchet distance over minimum homotopy area is that the Fréchet distance gives a metric on paths with different end points where minimum homotopy area does not. Consider two paths where one path is the reversal of the other. These curves will have a minimum homotopy area of zero but they are not equal. Moreover, if we form a closed curve from paths by connecting the end points, as described in Section 5.2, minimum

homotopy area does not satisfy the triangle inequality. Let P and Q be two non-equal parallel segments and let R be the point in center of the rectangle formed by P and Q . Then

$$\text{Area}_H(P, Q) > \text{Area}_H(P, R) + \text{Area}_H(R, Q).$$

However, if all of the curves we are considering have the same start and endpoints, then minimum homotopy area is a metric.

Another advantage of minimum homotopy area over the Fréchet distance is that homotopy area can be generalized to curves and paths on surfaces other than \mathbb{R}^2 in a natural way while the Fréchet distance does not take the underlying topology of a surface into account. Terrains are often modeled by triangulations embedded in \mathbb{R}^3 . Consider the two curves on the terrain given in Figure 5.6. The curves have a relatively small Fréchet distance in \mathbb{R}^3 and a relatively large Fréchet distance when the distances are restricted to the surface. To remedy this, Guibas et. al consider the *geodesic Fréchet distance* [31]. The geodesic Fréchet distance uses the Euclidean shortest path distance on the surface instead of the L_2 norm and can be applied to ovoid obstacles in the domain [31, 45, 50].

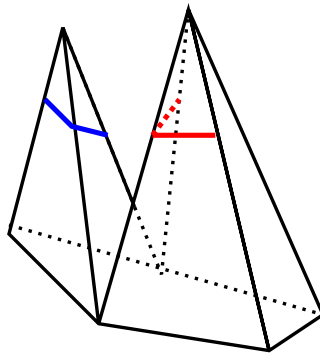


Figure 5.6: Two curves on a polygonal terrain with relatively small Fréchet distance in \mathbb{R}^3 and a relatively large geodesic Fréchet distance.

If we interpret the geodesic Fréchet distance as a leash, in the geodesic Fréchet distance we see that the leash would need to be ‘magic’ in sense that it would jump over obstacles. If

the leash cannot switch discontinuously from one position to another, the distance is called the *homotopic Fréchet distance*. Algorithms to compute the homotopic Fréchet distance are considered in [17, 43].

CHAPTER SIX

HOMOTOPY AREA ON SURFACES WITH POSITIVE GENUS

In this chapter, we consider computing optimal homotopies on orientable triangulated surfaces of positive genus without boundary. Past results focus on computing the minimum homotopy area between two homotopic simple paths that share endpoints [16]. Our algorithm extends this result in two ways. First, we compute the minimum homotopy area between non-simple homotopic paths that share endpoints, and second, we compute the minimum homotopy area between two non-intersecting, non-contractible homotopic closed curves.

A *surface* is a compact, connected topological space where each point has a neighborhood homeomorphic to an open disk. A *triangulation* of a surface is a representation as a union of triangles, edges, and vertices such that each pair of triangles either share a single vertex, a single edge, or are completely disjoint. The triangles incident to a vertex can be ordered circularly where two triangles are adjacent in the ordering if and only if they share an edge. The *genus* of a connected, orientable surface is the maximum number of cuttings along non-intersecting simple closed curves without disconnecting the surface [54].

We give a polynomial time algorithm to answer the following question:

Question 1. *Given two non-contractible, non-intersecting curves on a surface, determine if the curves are homotopic, and, if so, compute the minimum homotopy area between them.*

See Figure 6.1 for an illustration of Question 1.

As input, we have a surface S and a triangulation T with complexity N , and two curves P and Q with combined complexity $n = n_p + n_q$. The total number of self-intersections of P and Q is $|V|$. We assume T is stored in a DCEL data structure [22]. We also assume that the edges of P and Q are edges from the triangulation T . The worst-case runtime of

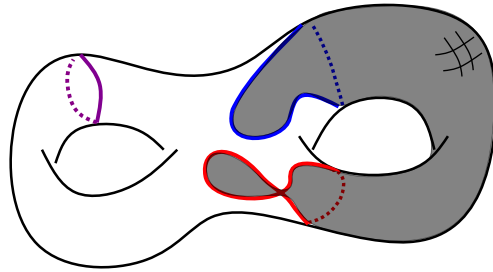


Figure 6.1: We see three curves on a surface of genus two. Two of the three curves are homotopic. Algorithm 6.2 computes the minimum homotopy area between them. The image of a minimum area homotopy is shaded.

our algorithm is

$$O(gN + ng + nN + n^2 + |V|^6).$$

We next describe how to combine our algorithm in the plane, Algorithm 5.1, with a lift to a covering space to compute the area of non-simple contractible curves on a surface. Then, we consider the homotopy area between two homotopic, non-contractible, non-intersecting curves on a surface.

6.1 The Universal Cover

Let S be a surface. A continuous function $p : \tilde{S} \rightarrow S$ is a *covering map* if for every point $x \in S$ there is an open neighborhood U such that $p^{-1}(U)$ is the union of disjoint open sets $\bigcup_i V_i$, and the restriction of p to each open set V_i is a homeomorphism [54]. If a surface S has a covering map $p : \tilde{S} \rightarrow S$ then \tilde{S} is called a *covering space* of S . The *universal cover* of S is the unique simply connected covering space of S . A *lift* of a path $\xi : [0, 1] \rightarrow S$ is a path $\tilde{\xi} : [0, 1] \rightarrow \tilde{S}$ such that $p \circ \tilde{\xi} = \xi$. A lift of a closed curve $\gamma : \mathbb{S}^1 \rightarrow S$ is a continuous function $\tilde{\gamma} : \mathbb{R} \rightarrow \tilde{S}$ such that for all t , $p(\tilde{\gamma}(t)) = \gamma(t \bmod 1)$.

We will use the following facts about universal covers and lifts.

Fact 1 (Specifying a Lift [65]). *Let ξ be a path in S with initial point x . If we pick a specific \tilde{x} in \tilde{S} with $p(\tilde{x}) = x$, then there is a unique path $\tilde{\xi}$ in \tilde{S} , starting at \tilde{x} with $p \circ \tilde{\xi} = \xi$.*

Fact 2 (Contractible if and only if Closed [17, 69]). *A closed curve γ in S is null-homotopic if and only if $\tilde{\gamma}$ is closed in \tilde{S} .*

Fact 3 (Intersections [16, 21]). *For a closed curve γ in S , and a particular lift $\tilde{\gamma}$ in \tilde{S} the number of self-intersections of γ is less than or equal to the number of self-intersections of $\tilde{\gamma}$.*

The universal cover of any orientable compact surface with $g > 0$ is homeomorphic to \mathbb{R}^2 . Chabers and Wang show that we can lift a curve in a surface to the universal cover and compute the minimum homotopy area of the lifted curve to obtain the minimum homotopy area of the original curve [16].

We canonically represent an orientable surface with genus $g > 0$ by a $4g$ -gon. This polygon is called a *polygonal schema* of the surface S [16, 27, 62, 69]. Each edge on the boundary of the polygonal schema has a signed label with each unsigned label occurring twice. Two edges with the same unsigned labels are *partnered edges*. See Figure 6.2 for an example. All of the $4g$ vertices represent the same vertex in S . Moreover, an orientable surface with genus g cannot be represented by a polygon with fewer than $4g$ edges [27]. The labels on the edges have a specific form, namely,

$$a_1 b_1 \bar{a}_1 \bar{b}_1 a_2 b_2 \bar{a}_2 \bar{b}_2 \dots a_g b_g \bar{a}_g \bar{b}_g. \quad (6.1)$$

After identification, the edges of the polygonal schema form $2g$ curves on S that are disjoint, apart from the common endpoint. These loops are the generators of the fundamental group and the first homology group of S [33, 48].

Given a surface we construct a polygonal schema in the following way. A *cut graph* is a set of curves that decompose a surface into a disk. A *system of loops* is a cut graph

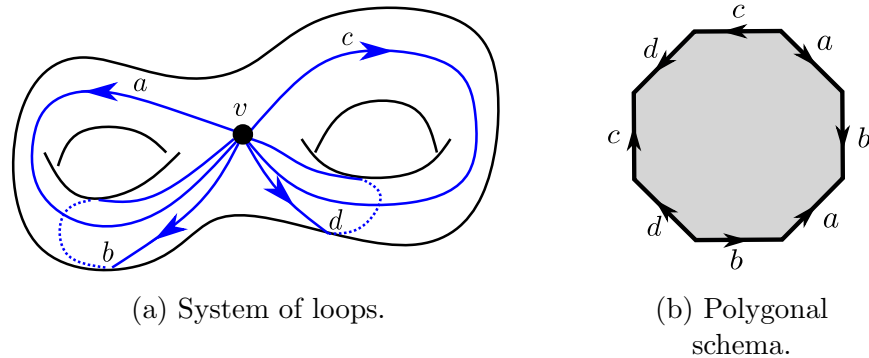


Figure 6.2: (a) A genus two surface with a system of loops. (b) The polygonal schema corresponding to the surface and the system of loops.

consisting of simple loops that share a common basepoint and are disjoint elsewhere; a *canonical system of loops* is a system of loops that meet in particular cyclic order around the base vertex. Algorithms to compute a canonical system of loops are considered in [21, 48, 69], and optimal-time algorithms run in

$$O(gN) \tag{6.2}$$

time in the worst-case.

Once we fix a basepoint for our system of loops, we introduce a difference between homotopy with a fixed basepoint and homotopy of cycles. We say two cycles in a homotopy class are *freely homotopic* when no basepoint is specified. Two cycles with a basepoint can be freely homotopic but fail to be homotopic when the basepoint is fixed. See Figure 6.3 for an example. Our algorithm computes the area between freely homotopic curves.

6.2 An Algorithm for Contractible Curves on a Surface

In this section, we give an algorithm to compute the minimum homotopy area of a contractible curve on a surface. Our strategy is to lift the curve to the universal cover and apply Algorithm 5.1.

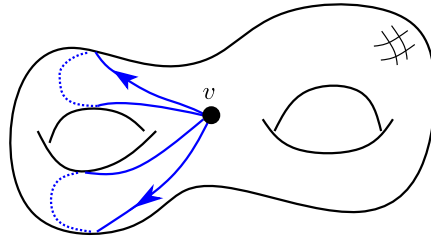


Figure 6.3: Two curves that are not homotopic when the basepoint is fixed but they are freely homotopic.

Theorem 7 (Homotopy Area of a Contractible Curve). *For a contractible closed curve in a surface with $g = 1$ we compute the minimum homotopy area in*

$$O(n^2N + |V|^6)$$

and for a contractible closed curve in a surface with $g > 1$ we compute the minimum homotopy area in

$$O(gN + ng + nN + (gn)^2 + |V|^6).$$

Proof. The universal cover consists of a infinite number of the polygonal schema glued together along identified edges. Each copy of the schema is a *tile*. In [27], Dey and Schipper give an algorithm to determine if a closed curve on a surface is contractible. Their algorithm constructs only a portion of the universal cover using one edge of the curve at a time. We use their algorithm to lift our curves. Let γ be a closed curve in S with complexity n that follows the edges of the triangles in T . By the Fact 1, once we choose a basepoint in $\tilde{\gamma}$ we can specific a lift. Dey and Schipper use Fact 2 to determine if γ is contractible by determining if the lifted curve $\tilde{\gamma}$ is closed.

The number of tiles bounded by the lifted curve depends on the genus. This leads to two cases. If $g = 1$, S is a torus and there can be $\Theta(n^2)$ interior tiles. However, if $g > 1$ then the number of interior tiles is $O(n)$ [27]. Dey and Schipper's algorithm has the following

runtimes to determine contractibility of a curve on a surface, for $g = 1$,

$$\Theta(N + n) \tag{6.3}$$

time and space and

$$O(N + n \log g) \tag{6.4}$$

time and $\Theta(N + n)$ space and when $g > 1$.

The output of their algorithm contains tiles that are not necessarily filled in with triangles from T . In order to compute the minimum homotopy area of γ , we need to consider all tiles in the region bounded by $\tilde{\gamma}$. Chambers and Wang describe how to adapt Dey and Schipper's algorithm to compute the optimal homotopy area. Since γ follows the edges of the triangles in S , when we cut S into a polygonal schema, each edge may be decomposed into $O(g)$ pieces. If γ consists of n edges, in the worst case, the complexity of our curve becomes

$$O(ng).$$

Once the curve is broken up into $O(gn)$ pieces, they compute the combinatorial structure of the lifted arrangement and the set of tiles in each face that the arrangement intersects or bounds.

For $g = 1$, the schema consists of rectangles with boundary $ab\bar{a}\bar{b}$. We assign the coordinate $(0, 0)$ to the tile that contains the lifted starting point. A tile t has coordinate (i, j) if the closed loops with homotopy type $a^i b^j$ starting at $(0, 0)$ end in t . This gives a sequence of tiles that the curve will pass through in

$$O(n + N)$$

time. Then, we compute the combinatorial structure of the lifted arrangement in

$$O(n + |V| \log(|V|)) \tag{6.5}$$

time without explicitly enumerating the $\Theta(n^2)$ tiles enclosed by the lifted curve. For $g > 1$ the runtime is bounded by

$$O(ng + |V| \log(|V|)). \tag{6.6}$$

Chambers and Wang then describe how to compute the area of the faces in the arrangement formed by the lifted curve. Their algorithm uses data structures that rely on the fact that their lifted curve consists of two simple paths. We do not make this assumption so we compute the areas of the cells in this arrangement in a straightforward manner. Let $\text{Arr}(\tilde{\gamma})$ denote the arrangement of the lifted curve. We first consider the case when $g = 1$. There are

$$\Theta(n^2)$$

tiles bounded by the lifted curve each containing $O(N)$ triangles giving a worst-case runtime of

$$O(n^2 N) \tag{6.7}$$

When $g > 1$, there are

$$\Theta(n)$$

tiles to consider each with $O(N)$ triangles and the runtime is bounded by

$$O(nN). \tag{6.8}$$

Thus, we prove the theorem by lifting the curve to the universal cover and applying Algorithm 5.1. By Fact 3, the lifted curve has at most the same number of self-intersections

as the original curve, so the runtime of Algorithm 5.1 applies. \square

We emphasize that our algorithm does not require the curves to be composed of simple homotopic paths that share endpoints. We next describe how we can extend this idea to compute the optimal homotopy area between two non-contractible, non-intersecting, homotopic curves.

6.3 Connector Between Homotopic Closed Curves

Given two non-contractible, non-intersecting, homotopic curves, γ_1 and γ_2 , we now show that we can lift them to a closed curve in the universal cover such that the minimum homotopy area of the lifted curve is equal to the minimum homotopy area between γ_1 and γ_2 in the original surface. The curves γ_1 and γ_2 may have self-intersections but we require that they do not intersect each other.

We will show that, since γ_1 and γ_2 are non-contractible, non-intersecting, and homotopic, we are able to connect them with a path β so that the curve $C = \gamma_1 \circ \beta \circ \text{rev}(\gamma_2) \circ \text{rev}(\beta)$ is null-homotopic with minimum area homotopy equal to the minimum area homotopy between γ_1 and γ_2 . We call the path β a *connector*. We cannot connect our two curves with any path as the resulting curve might not be contractible, an example is given in Figure 6.4.

The curves γ_1 and γ_2 are non-contractible, non-intersecting, and homotopic by assumption. Any homotopy between them determines an annulus. On a surface with $g = 1$, there are two possible annuli. One that is swept out by a minimum area homotopy and the other is the complement. We consider connecting the curves in both annuli and take the minimum. When $g > 1$, we use the Euler characteristic to determine which component of the complement of the curves is an annulus and use this subsurface to construct our connector.

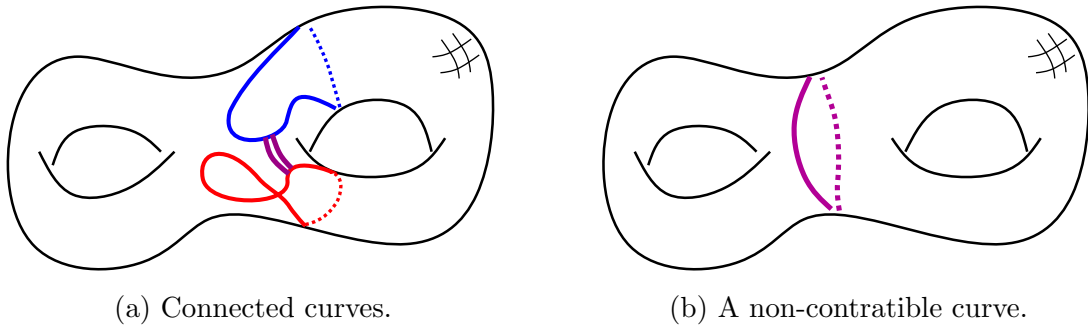


Figure 6.4: An example illustrating that care must be taken in how we connect the curves. (a) Two homotopic curves on a surface connected by a path. The resulting curve is in the homotopy class shown in (b) and is not contractible.

Lemma 8 (Connector Construction). *Let γ_1 and γ_2 be two non-contractible, non-intersecting, homotopic, closed curves in a surface S . In $O(N)$ time, we construct a path β such that the curve $C = \gamma_1 \circ \beta \circ \text{rev}(\gamma_2) \circ \text{rev}(\beta)$ is null-homotopic.*

Proof. We decompose the surface using γ_1 and γ_2 . Since the curves might not be simple, there may be several connected components after the decomposition. We pick a starting point not on γ_1 or γ_2 and breadth-first search until we obtain a bounded subsurface. While searching, we maintain the Euler characteristic. Each subsurface S' is connected compact orientable with boundary, thus, the Euler characteristic is given by

$$\chi(S') = 2 - 2g - b.$$

The Euler characteristic of an orientable subsurface uniquely determines topology of the subsurface we determine which subsurface is an annulus by determining which subsurface has Euler characteristic zero. The annulus might be knotted but the Euler characteristic does not change. We traverse the surface once exactly once, in $O(N)$ time, we identify the annulus and find a shortest path.

Since γ_1 and γ_2 do not intersect each other, there is one annulus with γ_1 on one boundary

and γ_2 on the other boundary. We pick a point on each boundary component and connect them with a shortest path. This shortest path is our connector β . The curve β is a cut in the annular domain of a homotopy, since any homotopy is continuous, the curve $C = \gamma_1 \circ \beta \circ \text{rev}(\gamma_2) \circ \text{rev}(\beta)$ is null-homotopic. \square

Slicing the image of a homotopy along a path does not change the area.

Theorem 8 (Lifted Area). *The minimum homotopy area of C in Lemma 8 is equal to the minimum homotopy area between γ_1 and γ_2 .*

Proof. Given any homotopy between γ_1 and γ_2 we obtain a null-homotopy of C by first performing the homotopy from γ_1 to γ_2 then contract curve $\gamma_2 \circ \text{rev}(\gamma_2)$ without adding area. Thus, the minimum homotopy area of C is less than or equal to the minimum homotopy area between γ_1 and γ_2 .

In the other direction, given any null-homotopy of C we construct a homotopy between γ_1 and γ_2 . Let x denote the point where β meets γ_1 and let y denote the point where β intersects γ_2 . We first extend a spike from x on γ_1 that follows β and then follows γ_2 . Adding a spike does not add any area. We then temporarily identify the point of the added spike with one of the strands of the spike at the point y . With this identification, the curve γ_1 with the added spike gives copy of C with a copy of γ_2 . Given any null-homotopy of C , we follow it with our copy of C and extend a spike from y if necessary. When the null-homotopy of C is finished, we have a copy of γ_2 with a spike from y to the point that C contracted to. Finally, we contract the spike to obtain a homotopy from γ_1 to γ_2 with area equal to the area of the null-homotopy of C . See Figure 6.5 and Figure 6.6 for an illustration. Thus, the minimum homotopy area between γ_1 and γ_2 is less than or equal to the minimum homotopy area of C . And we have proved the corollary. \square

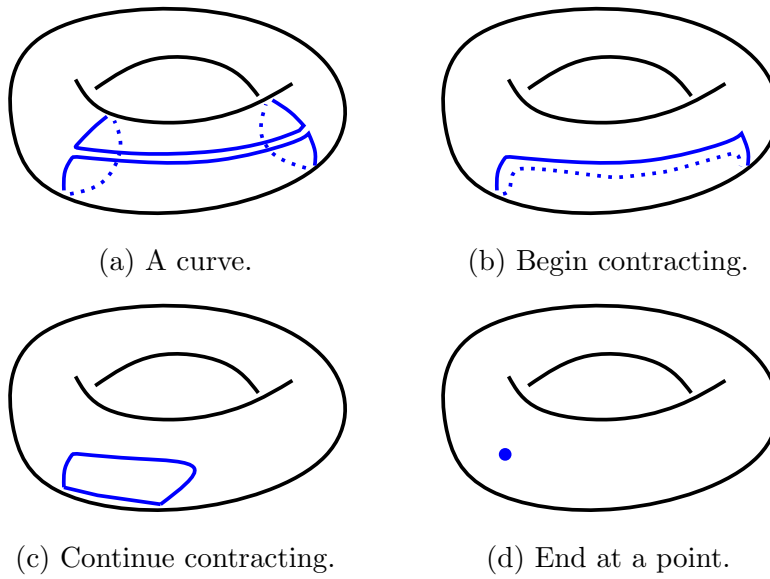


Figure 6.5: An illustration of a null-homotopy of a curve of the form $\gamma_1 \circ \beta \circ \text{rev}(\gamma_2) \circ \text{rev}(\beta)$. The area of the homotopy is equal to the area of the homotopy in Figure 6.6. (a) A closed curve C drawn in blue on a surface. In Figure 6.6, we construct a homotopy from γ_1 to γ_2 with equal area. (b) A null-homotopy of the curve in progress. (c) The homotopy continues. (d) The null-homotopy ends at a point.

6.4 Putting the Pieces Together

We now bound the run time of Algorithm 6.2. Let n denote the total complexity of both curves. First, we determine if our curves γ_1 and γ_2 are homotopic in

$$O(N + n)$$

time using an algorithm of Dey and Guha [26]. In Line 5, we build our connector in $O(N)$ time using Lemma 8.

We construct our system of loops in

$$O(gN)$$

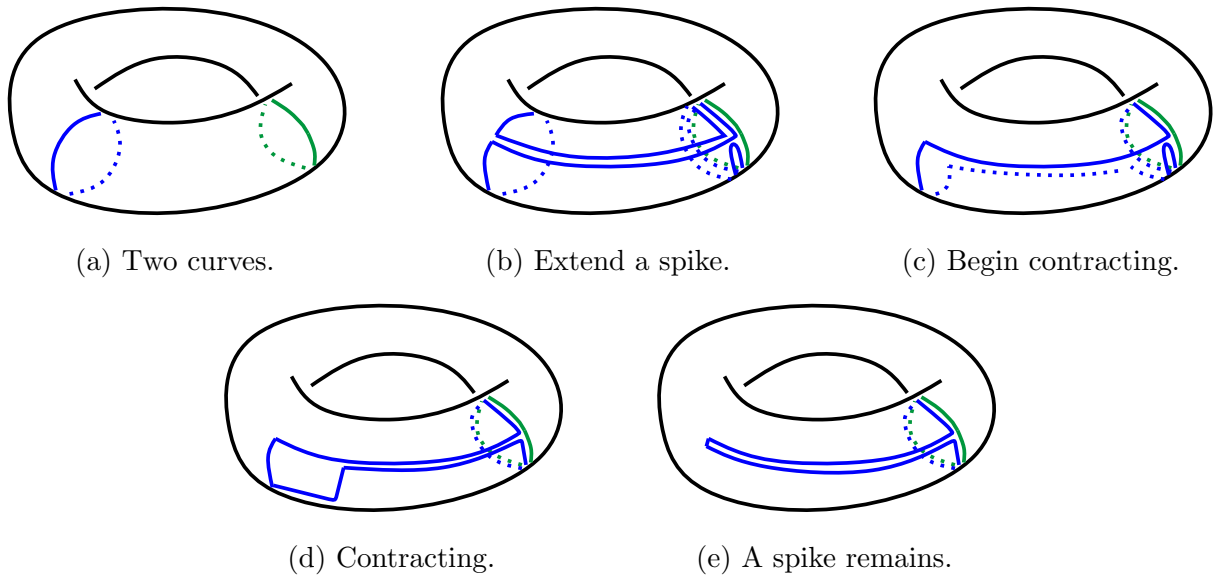


Figure 6.6: An illustration of a homotopy between two curves that has equal area as the homotopy of the curve in Figure 6.5. (a) Two homotopic curves γ_1 and γ_2 on a surface. For any homotopy of C we construct a homotopy from γ_1 to γ_2 that sweeps out equal area. (b) We add a spike from γ_1 that follows the connector of C then follows γ_2 . (c) We now contract the copy of C using the homotopy of C in Figure 6.5. (d) We continue contracting. (e) We are left with a copy of γ_2 with an additional spike. Thus, we have a homotopy from γ_1 to γ_2 with area equal to the area of C .

using Equation 6.2. We compute the relevant portion of the universal cover, lift our curve, and compute areas of the faces in the lifted arrangement

$$O(n^2N + |V| \log(|V|))$$

when $g = 1$ and

$$O(ng + nN + |V| \log(|V|))$$

when $g > 1$ using Equation 6.6 and Equation 6.5 respectively.

By Fact 3, there are fewer self-intersections on the lifted curve. We then compute the

minimum homotopy area of the lifted curve in

$$O(n^2 + |F|^6)$$

time using Algorithm 5.1. Thus, we answer Question 1 with the following. Pseudocode for our algorithm is given in Algorithm 6.2.

Theorem 9 (Minimum Homotopy Area Between Two Curves). *For two non-contractible curves in a surface with $g = 1$, we compute the minimum homotopy area between them in*

$$O(n^2N + |V|^6) \tag{6.9}$$

and, for a two non-contractible curves in a surface with $g > 1$, we compute the minimum homotopy area between them in

$$O(gN + ng + nN + (gn)^2 + |V|^6). \tag{6.10}$$

in the worst-case.

6.5 Discussion

In the plane, a point is the shortest curve homotopic to any given curve. Our algorithm in the plane computes the minimum homotopy area between a curve and the shortest homotopic curve. We can compute the same quantity on surfaces provided the shortest homotopic curve is unique. In [20], de Verdière and Erickson show that, for a surface with genus $g > 1$, we can compute the shortest cycle homotopic to a given cycle with complexity n in

$$O(gNn \log(Nn)) \tag{6.11}$$

Algorithm 6.2: `MinimumSurfaceHomotopy`(S, γ_1, γ_2)

Input: A surface S and two homotopic, non-contractible, non-intersecting curves γ_1 and γ_2 .

Output: If γ_1 and γ_2 are homotopic, return the minimum homotopy area, otherwise return that they are not homotopic.

- 1: **if** γ_1 and γ_2 are not homotopic **then**
- 2: **return** Not homotopic
- 3: **end if**
- 4: Construct a system of loops
- 5: Construct a connector from γ_1 to γ_2 to obtain a contractible closed curve
- 6: Lift the curve to the covering space
- 7: Compute areas of the faces of the lifted curve
- 8: Compute minimum homotopy area using Algorithm 5.1
- 9: **return** The minimum homotopy area between γ_1 and γ_2 .

after $O(N^2 \log N)$ preprocessing. Thus, we can use Algorithm 6.2, to find the minimum homotopy area between a given curve and the shortest curve homotopic to our given curve.

Computing the minimum homotopy area on the sphere remains an interesting open problem. Since the sphere is simply connected all curves are contractible and the universal cover of the sphere is itself. Thus, we cannot lift to a space homeomorphic to \mathbb{R}^2 as we did for surfaces with $g \geq 1$. If we know that a face on the sphere is now swept over by a minimum area homotopy, we can use stereographic projection to apply our algorithm in the plane. However, there exist curves such that any minimum area homotopy sweeps over the entire sphere.

CHAPTER SEVEN

FROM WORDS TO CURVES

We now consider the problem of determining which Blank words correspond to curves in the plane. See Figure 7.1a for an example of a Blank word that corresponds to a curve, and see Figure 7.1b for a word that does not correspond to a curve. We are motivated to consider this problem because it is similar to the historic problem of determining which Gauss codes correspond to curves [23, 24, 29, 42, 55, 61]. Recall, to construct the Gauss code from a curve, assign each crossing a label, then traverse the curve and record the sequence of labels encountered. See Figure 7.2a for an example of a Gauss code that corresponds to a curve and see Figure 7.2b for a Gauss code that does not correspond to a curve. An algorithm to determine if a Gauss code corresponds to a curve that is quadratic in the number of self-intersections is given in [35]. Erickson suggests that determining whether a Gauss code corresponds to an actual plane curve is one of the earliest computational topology questions [35].

In this chapter, we give an algorithm that solves a special case of this problem where each letter appears exactly once in the word. Our algorithm is linear in the length of the word. We then give a quadratic algorithm to determine if a combined word corresponds to a closed curve in the plane. Finally, we use our algorithm for the combined word to give an exponential algorithm to determine if a Blank word corresponds to a closed curve.

Given a Blank word, a combined word, or a Gauss code, we store the string of length \mathcal{L} in a circularly linked list. In $O(\mathcal{L})$ time, we add a pointer to each vertex symbol that gives us the other occurrence of the vertex in the word [35]. See Figure 7.3 for an example. Thus, given a vertex symbol in a word v we have a function, `match[v]`, that gives the other occurrence of v in constant time.

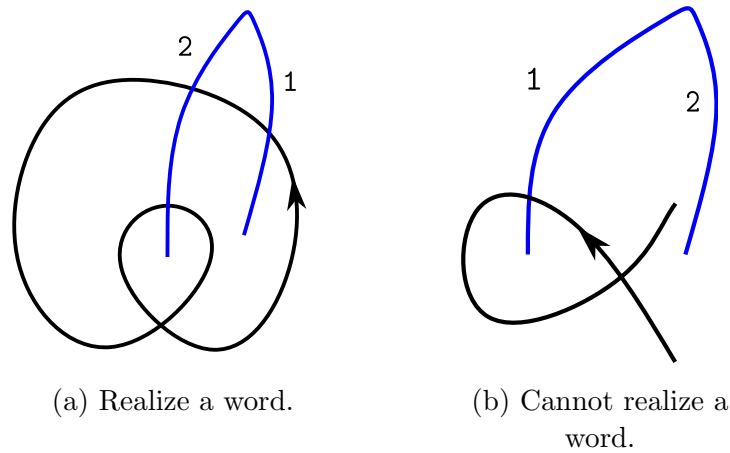


Figure 7.1: Some Blank words correspond to curves and others do not. (a) There does exist a curve with Blank word $[122]$. (b) There does not exist a curve with Blank word $[12]$.

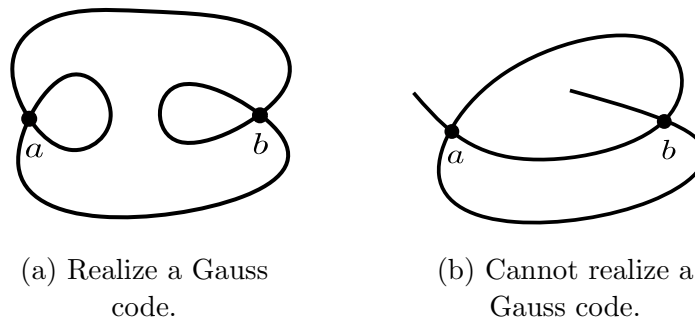


Figure 7.2: Some Gauss codes correspond to curves and others do not. (a) There does exist a curve with Gauss code $[aabb]$. (b) There does not exist a curve with Gauss code $[abab]$.

7.1 A Special Case

In this section, we solve a special case of our inverse problem. If every face in a curve has depth one, we call the curve a *chain*; see Figure 7.4 for an illustration. We give an algorithm to determine if a Blank word corresponds to a chain. The algorithm is linear in the length of the word.

A *monogon* is a face in a curve bounded by a single edge. See Figure 7.4 for an example

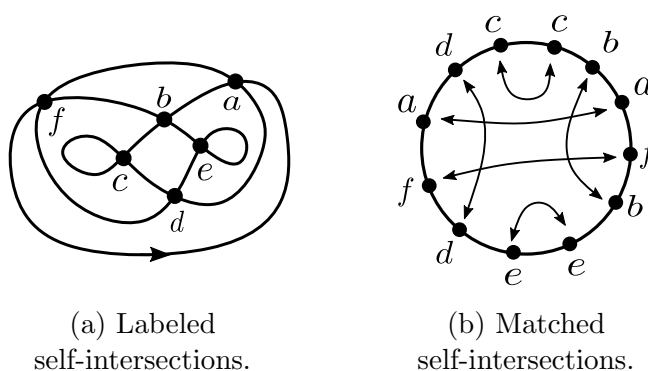


Figure 7.3: In linear time, we add pointers from each vertex to the other occurrence of the vertex. In (a) we have a curve with labeled vertices and in (b) the Gauss diagram for the curve.

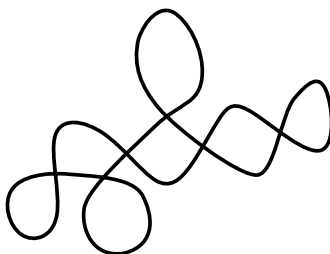


Figure 7.4: A curve where each face has depth one. There are four monogons.

of a curve with monogons. Recall from Section 4.3, a vertex is said to be sign-changing if the four incident faces have winding numbers $[1, 0, -1, 0]$ in cyclic order. Since each letter appears exactly once, every vertex is a sign-changing vertex and it connects two faces with opposite orientations. If a word consists of a single symbol, then the word trivially corresponds to a simple curve. Otherwise, the word has more than one letter and we show that the word corresponds to a curve if and only if there are letters with opposite signs in the word.

Theorem 10 (Depth One Inverse). *Let w be a word with length greater than one, such that each letter appears exactly once. Then, w corresponds to a curve if and only if there are both positive and negative letters in the word.*

Proof. By assumption, w contains more than one letter. The corresponding curve must have more than one face and at least one vertex. If all letters have the same sign then w cannot correspond to a curve because each vertex must be adjacent to a positive and a negative face.

Conversely, if w has more than one letter and there is at least one letter with each sign, we give an algorithm to construct a drawing of a curve that corresponds to w . First, iterate over the word until we find a pair of adjacent letters where the first letter is positive and the second letter is negative. We construct a figure eight with two cables that corresponds to this two letter subword. Let v_0 denote the single vertex and let $\gamma(0) = \gamma(1)$ be a point between v_0 and the intersection between cable and the subcurve with positive orientation. See Figure 7.5a for an example. Let $p_1 \in [0, 1]$ be the point such that $\gamma(p_1)$ intersects the the cable with negative orientation.

We call the path $\gamma(t)$ for $t \in [p_1, 1]$ the *attachment segment* of the curve. The attachment segment consists of a subpath $[p_1, v_0]$ with a negative orientation and subpath $[v_0, 1]$ with a positive orientation. We now consider the next letter in the word ℓ . If ℓ is positive, we attach a monogon at a new vertex v_1 to the negative part of the attachment segment and add a cable. Call the point where the curve intersects the new cable p_2 and update the attachment segment to have a positive interval from $[p_2, v_1]$, a negative interval $[v_1, v_0]$ and another positive interval $[v_0, 1]$. If the ℓ is negative, attach a monogon at a new vertex v_1 on the positive part of the tail and call the point where the new cable intersects the curve p_2 . Then, update the attachment segment to be $[p_2, v_1]$ with negative orientation and $[v_1, 1]$ with positive orientation.

Continue adding monogons in this fashion, the property that we maintain is that the attachment segment always contains subpaths with positive and negative orientation and we can attach additional faces as needed. After we have processed all letters in w , we have a curve that corresponds to the Blank word. \square

An example of the above construction is given in Figure 7.5. We iterate over the word

twice, once to find adjacent letters with opposite sign and once to process the remaining letters. Thus, in linear time, we determine if a Blank word with each letter appearing exactly once corresponds to a closed curve in the plane.

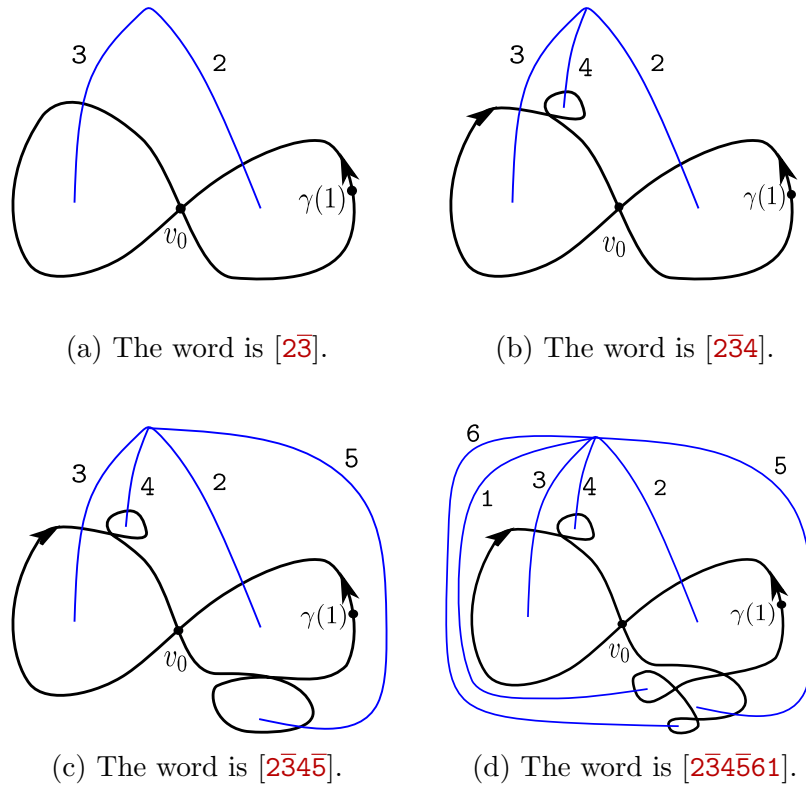


Figure 7.5: We are given the Blank word $w = [12\bar{3}4\bar{5}6]$. Since w has positive and negative letters, we construct a curve that corresponds to w . (a) A curve that corresponds to two letters with opposite sign, $[2\bar{3}]$. (b) We add the next letter, $[2\bar{3}4]$. It is positive, so we add it to the negative part of the attachment segment. (c) We add the next letter, $[2\bar{3}4\bar{5}]$. It is negative, so we add it to the positive part of the attachment segment. (d) The next two letters are positive, $[2\bar{3}4\bar{5}61]$. So we add them to the negative part of the attachment segment and we have constructed a curve that corresponds to w .

7.2 Combined Words to Curves

Recall, the combined word includes both the Blank word and the Gauss code. We now determine if a combined words correspond to curve. We will use this algorithm as a

subroutine to determine if a Blank word corresponds to a curve. Our algorithm will trace out the boundary of each face in the curve. This face tracing strategy is used by Carter in [15] to determine which signed Gauss codes correspond to curves. We now give a brief description of Carter's algorithm.

The signed Gauss code contains additional crossing information. Consider a point traversing the curve in a chosen direction, a crossing is positive if the point crosses the curve from right to left and negative if we cross the point from left to right. See Figure 7.6 for an illustration of a signed crossing. The curve in Figure 7.2b has signed Gauss code $[a^+a^-b^+b^-]$.

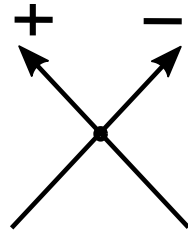


Figure 7.6: The crossing from right to left is positive and the crossing from left to right is negative.

Carter gave algorithm to determine if a signed Gauss code corresponds to a curve that is linear in the length of the word [15, 35]. The algorithm traces out the faces of the curve. To see this, imagine a point moving counterclockwise around a bounded face. When we encounter a vertex we determine if we turn left based on the following cases:

- Encounter a positive vertex v^+ forward, we leave $v^- = \text{match}[v^+]$ backwards.
- Encounter a negative vertex v^- forward, we leave $v^+ = \text{match}[v^-]$ forward.
- Encounter a positive vertex v^+ backward, we leave $v^- = \text{match}[v^+]$ forward.
- Encounter a negative vertex v^- backward, we leave $v^+ = \text{match}[v^-]$ backward.

After tracing each face in the curve, we then determine if we can stitch the faces together to form a closed curve.

For example, consider the signed Gauss code $[a^+a^-b^+b^-]$. The curve is shown in Figure 7.2b. We first encounter a^+ , we then leave from a^- in the reverse direction and encounter a^+ again. We are now in the third point above, we leave a^- forward reaching b^+ . We leave b^- backwards hitting b^+ again. We leave b^- forward and we have traced out the face with depth one. This process is repeated, starting with each vertex, until we have traced out all faces.

The combined word includes both the Blank word and the Gauss code, Section 4.2. If there are $2|V|$ vertex symbols in the combined word there can be $O(|V|^2)$ face symbols, see Figure 5.1 for an example.

Combined words are given to us as a string of length \mathcal{L} stored in a circularly linked list. We have the function `match[v]`, that gives the other occurrence of v in constant time after $O(\mathcal{L})$ preprocessing. The depth of each face is encoded in the Blank word.

Remark 1. *Because the cables follow shortest paths to the unbounded face, the number of instances of a face in the Blank word is equal to the depth of the face in the curve.*

By iterating over the word once, we record the number of occurrences of each face symbol and thus the depth of each face in $O(\mathcal{L})$ time. If there is a face with depth d and no faces with depth $d - 1$ we can conclude that the word does not correspond to a curve. We also use Euler's formula to determine if some combined words do not correspond to a curve.

Remark 2. *Because a closed curve can be thought of as a four-regular graph in the plane, the number of edges is twice the number of vertices. By Euler's theorem, the number of faces in the combined word is one less than the number of vertices. If this is not the case, the curve is not planar.*

We now include some simple observations that help us determine if a combined word corresponds to a curve.

Lemma 9 (Gauss Code Elimination). *Given a combined word, if we forget the faces and the resulting Gauss code does not correspond to a planar curve, then the combined word does not correspond to a planar curve.*

Proof. If the vertices do not correspond to a curve, adding cables will not create a curve. \square

We can identify monogons in the combined word.

Lemma 10 (Vertex Face Vertex Monogon). *Let w be a combined word and let v be a vertex and f a face in the word. Then w has a subsequence of the form $\dots vfv\dots$ if and only if f is a monogon.*

Proof. Given a subsequence of the form $\dots vfv\dots$, the repeated vertex forms form a Jordan curve with no other vertices. Thus, we have a single face with cable f . Conversely, if we have a monogon, we have a single edge bounding a face and we have a subsequence of the form $\dots vfv\dots$ \square

As a preprocessing step, we remove any monogons from a given word.

Lemma 11 (Remove Monogons). *Given a combined word w , let w' denote the word obtained by deleting all monogons from w . Then, w corresponds to a curve if and only if w' does.*

Proof. By Lemma 10, we can identify faces in the word that are monogons. When a monogon is removed, we obtain w' from w by removing the subsequence of the form vfv and all other occurrences of f . Monogons do not intersect the curve other than their single vertex, so, removing a monogon does not change the curve outside of the single vertex. And w corresponds to a curve if and only if w' does. \square

Lemma 11 implies we only need to consider curves that do not contain monogons and for the remainder of this chapter, we assume that our curves do not contain monogons. The following lemma enables us to trace the boundary of each face in a combined word. We require all cables to be shortest paths and reduced.

Lemma 12 (Unique Simple Cycle Per Face). *If a combined word w corresponds to a curve, then there is a unique simple cycle such that the cable corresponding to a face intersects the cycle exactly once and no other cable intersects the cycle exactly once.*

Proof. If a combined word corresponds to a curve, each face has a unique cable emanating from a point in the interior of a face. By the Jordan curve theorem, this cable intersects the boundary of the face exactly once. Because all cables to be shortest paths and reduced, the boundary of every other face intersects this cable twice or does not intersect the cable. \square

Given a combined word w of length \mathcal{L} , we now give a $O(\mathcal{L}^2)$ algorithm to determine if a combined word corresponds to a curve. First, we trace the boundary of a face with depth one in $O(\mathcal{L})$ time. Then, we trace the boundary of all faces that have a cable that pass through the face of depth one before considering the next face of depth one. Finally, we determine if we can glue the boundaries together to form a curve.

We now describe how to trace out the boundary of each face. An example is given in Figure 7.7. Every letter that appears exactly once corresponds to a face of depth one and the sign of the letter tells us the orientation of the curve. Thus, for a face of depth one f , we have an oriented edge on the exterior of the curve call the tail of this edge t and the head h . The sequence $t \dots f \dots h$ is a subword, where there are possible additional cables between t and f and between f and h . If $t = h$, then there is single face of depth one and we remove the substring and the vertex $h = t$ and start over. Otherwise, $t \neq h$, and we backtrack from the tail t of this edge and populate a binary tree T rooted at t . Each node in T is a potential boundary of the face, and we determine if it is the boundary by applying Lemma 12. Since all monogons have been removed, the boundary of f contains the vertex t and one of the two edges that intersect t transverse to the edge $t \dots f \dots h$. Since we do not know the orientation of the curve that intersects t , we consider both.

We next consider the substrings between the vertices to the left and to the right of

$\text{match}[t]$, call them a and b . Now, we have a subword of the form $a \dots t \dots b$ with possible cables in between. We prepend the substrings $a \dots t$ and $b \dots t$, with any cables in between, as the children of t in T . If either a or b equals h , we check if the cycle contains f and no other faces exactly once, if this is the case, then we have found the boundary of f by Lemma 12. Otherwise, we continue adding leaves to T by considering the neighbors of $\text{match}[a]$ and $\text{match}[b]$.

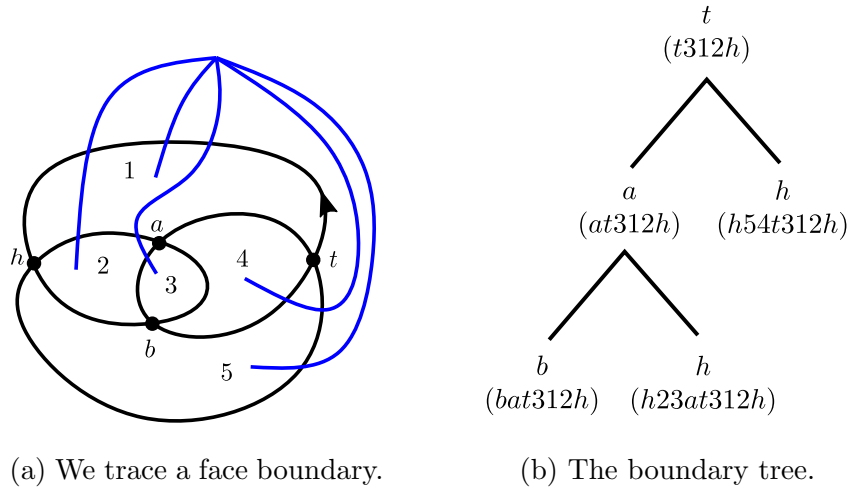


Figure 7.7: An example of tracing out a face of depth one. (a) A curve with combined word $w = [\mathbf{t}312\mathbf{h}b\mathbf{a}32\mathbf{h}54\mathbf{t}a3\mathbf{b}4]$. We wish to find the boundary of the face with label one. The initial edge, and root of our tree, is $t312h$. (b) The vertices adjacent to $\text{match}[t]$ give the substrings a and $h54$, we prepend them to the current string to become the children of the root. Since we reach h , we make a leaf, we then check if face one is the only face that appears exactly once. It is not, so we have not found the boundary. We then consider $\text{match}[a]$. We find h and this time, face one is the only face that appears exactly once, and we have found the boundary of face one.

We next bound the number of nodes in the binary tree T .

Lemma 13 (Bound on Binary Tree). *Each node in the tree corresponds to a unique edge in the potential curve and there are $O(\mathcal{L})$ nodes in the tree.*

Proof. Each substring prepended in a node in the tree begins and ends at a vertex. This corresponds to an edge in the potential curve. We mark each vertex we encounter as explored when it is prepended to the substring and so we do leave from a given vertex more than

once. Thus, each node in the tree corresponds to a unique edge. By Euler's formula, the number of edges in a potential curve is less than the length of the combined word, so there are $O(\mathcal{L})$ nodes in the tree. \square

Since the boundary of each face is a simple cycle, if we return to a vertex other than h we have found a cycle that is not the boundary of the face and we have a leaf in the tree. Pseudocode for tracing the boundary of a face is given below in Algorithm 7.3.

Algorithm 7.3: **FaceTrace**(w, f, e)

Input: A combined word, orientation, face f , exterior edge of the face e

Output: The boundary of the face

- 1: Let T be a binary tree with root t containing $\partial = [e] = [t \dots f \dots h]$
- 2: Mark h and t explored, let $c = t$ be the current node in T , let c' denote $\text{match}[c]$
- 3: **while** T has an unexplored node **do**
- 4: **if** The left vertex neighbor ℓ of c' is explored **then**
- 5: **if** $\ell = h$ **then**
- 6: Let u denote the subword $\ell \dots c'$, prepend u to ∂
- 7: **if** f is the only face that appears once in ∂ **then return** ∂
- 8: **end if**
- 9: **end if**
- 10: **else**
- 11: Set ℓ as explored and the left child of c to have $\partial = [u\partial]$
- 12: **end if**
- 13: Repeat the above process on the right vertex neighbor r of c' .
- 14: $c \rightarrow$ next node in the tree.
- 15: **end while**

For each letter that corresponds to a face of depth one, we determine if there are any

cables that pass through this face to deeper faces. These cables are between the same vertices as the face of depth one in the combined word. If there are, we add them to a queue, and compute their boundary in order of depth. In the example given in Figure 7.7, we have the edge $t312h$ where face one has depth one. We enqueue face two and face three. We computed the boundary of face one, to be $23at312h$. We call `FaceTrace[w, 2, a32h]` to compute the boundary of face two. The boundary of face two is $b3a32h$, we then call `FaceTrace[w, 3, a3b]` to compute the boundary of face three to be $a3b$. Pseudocode for this process is given below in Algorithm 7.4.

Algorithm 7.4: `DrillDown(w, f, e)`

Input: Combined word, face f of depth one, exterior edge of the face $e = t \dots f \dots h$

Output: The boundary of all faces in the subword $e = t \dots f \dots h$

- 1: Enqueue all faces with cables that intersect the edge $t \dots f \dots h$ in order of their depth in Q .
- 2: Let the initial exterior edge be $e = t \dots f \dots h$, and enqueue face f .
- 3: **while** $Q \neq \emptyset$ **do**
- 4: `FaceTrace[w, q, e]`
- 5: Recorded the boundary of face q
- 6: Iterate around the boundary of face q to find the exterior edge, update e .
- 7: **end while**
- 8: Combine boundaries.
- 9: **return** The boundary of all faces in the subword $t \dots f \dots h$

We then combine the boundary of the faces that have cables that pass through a given face of depth one. In the example given in Figure 7.7, the boundary of the union of the three faces that have a cable that passes through face one is $bat321h$. Once we have computed the boundary of all faces that pass through a face with depth one, we proceed to the next face of depth one by letting $t = \text{match}[h]$.

Given two cables of depth one with no other depth one cables between them, we check if we can glue the two boundaries together by matching substrings of the boundaries in $O(\mathcal{L})$ time. Once this process is complete, we iterate around the curve and check that the curve generates the word. If it does, we have found a curve with the given combined word. If it does not, we conclude that no curve exists that produces the given word.

Now to justify the $O(\mathcal{L}^2)$ runtime. We find all faces with depth one in $O(\mathcal{L})$ time by iterating over the combined word once. For each face of depth one, we call Algorithm 7.3 on all cables that pass through this face. By Lemma 13, Algorithm 7.3 builds a tree with $O(\mathcal{L})$. At each node in the tree we determine if a face appears once to apply Lemma 12.

Lemma 14 (Count Faces in Substrings). *We determine if a face appears once in substring in $O(\mathcal{L})$ time.*

Proof. Let A be an array of length \mathcal{L} that stores the number of times each face appears in the subword, we also maintain the number of faces that appear once in the A we denote this number r . When we encounter a new instance of a face in a subword, we update A , if that face now appears more than once, we decrease r by one, if the face previously did not appear in the subword we increase r by one. By Lemma 13, each node in T corresponds to an edge in the curve, we consider adding each symbol in the word to A once. Thus, we do $O(\mathcal{L})$ work to maintain A as we iterate over T and can check if $r = 1$ to determine if the face of interest is the only face that appears once in the subword. \square

Thus, by Lemma 13 and Lemma 14, Algorithm 7.3 takes $O(\mathcal{L})$ time and we call this on each face and our total runtime to determine if a combined word corresponds to a closed curve in the plane is $O(\mathcal{L}^2)$.

7.3 Blank Words to Curves

In this section, we use Algorithm 7.3 as a subroutine to determine if a general Blank word corresponds to a curve in exponential time. Given a Blank word, we try all possible ways to insert vertices into the word and determine if combined word corresponds to a curve. If one combined word does, then the Blank word does correspond to a curve. Otherwise, no curve exists. Let the length of the word be \mathcal{L} and let $|F|$ denote the number of distinct face letters, then \mathcal{L} is $O(|F|^2)$.

Theorem 11 (Blank Words to Curves). *Given a Blank word w of length \mathcal{L} , we determine if w corresponds to a curve in $O(\mathcal{L}^{2|F|})$.*

Proof. By Euler's formula, there are $|F| - 1$ vertices, each of which appears twice in the combined word. For each $2|F| - 2$ vertex symbols we have \mathcal{L} spaces between the face symbols in w where they could possible be placed. We use Algorithm 7.3 to test if each combined word corresponds to a curve in $O(\mathcal{L}^2)$ time. If one of the combined words does correspond to a curve, we forget the vertices and the curve realizes the given Blank word. Otherwise, no curve exists. □

We conjecture that a polynomial algorithm exists to determine if a Blank word corresponds to a curve and we hope that others consider this problem.

REFERENCES CITED

- [1] Pankaj Agarwal, Rinat Ben Avraham, Haim Kaplan, and Micha Sharir. Computing the Discrete Fréchet Distance in Subquadratic Time. *SIAM J. Comput.*, 43(2):429–449, 2014.
- [2] Helmut Alt and Michael Godau. Computing the fréchet distance between two polygonal curves. *Internat. J. of Comput. Geom. Appl.*, 5:75–91, 1995.
- [3] Helmut Alt, Bernd Behrends, and Johannes Blömer. Approximate matching of polygonal shapes. *Ann. of Math. and Artif. Intell.*, 13(3-4):251–265, 1995.
- [4] Helmut Alt, Christian Knauer, and Carola Wenk. Matching Polygonal Curves with Respect to the Fréchet Distance. volume 2010, pages 63–74. 18th Ann. Symp. Theo. Asp. Comput. Sci., 2001.
- [5] Helmut Alt, Christian Knauer, and Carola Wenk. Comparison of Distance Measures for Planar Curves. *Algorithmica*, 38(1):45–58, 2004.
- [6] Samuel Joel Blank. *Extending Immersions and Regular Homotopies in Codimension 1*. PhD thesis, Brandeis University, May 1967.
- [7] Bart Braden. The surveyor’s area formula. *The College Mathematics Journal*, 17(4):326–337, 1986.
- [8] Michael Brandenbursky, Światosław Gal, Jarek Kędra, and Michall Marcinkowski. The cancelation norm and the geometry of bi-invariant word metrics. *Glasg. Math. J.*, page 153–176, 2015.
- [9] Karl Bringmann. Why Walking the Dog Takes Time: Frechet Distance Has No Strongly Subquadratic Algorithms Unless SETH Fails. In *IEEE 55th Ann. Symp. on Found. of Comput. Sci.*, pages 661–670, 2014.
- [10] Karl Bringmann, Fabrizio Grandoni, Barna Saha, and Virginia Vassilevska Williams. Truly subcubic algorithms for language edit distance and RNA folding via fast bounded-difference min-plus product. *SIAM J. Comput.*, 48(2):481–512, 2019.
- [11] Karl Bringmann, Marvin Künnemann, and André Nusser. Walking the Dog Fast in Practice: Algorithm Engineering of the Fréchet Distance. *35th Symp. Comput. Geom.*, 2019.
- [12] Kevin Buchin, Maïke Buchin, Christian Knauer, Günter Rote, and Carola Wenk. How difficult is it to walk the dog? In *23rd european workshop on comput. Geom.*, pages 170–173, 2007.

- [13] Kevin Buchin, Maike Buchin, and André Schulz. Fréchet Distance of Surfaces: Some Simple Hard Cases. In *Algorithms – ESA 2010*, volume 6347, pages 63–74. Springer Berlin Heidelberg, 2010.
- [14] Kevin Buchin, Tim Ophelders, and Bettina Speckmann. Computing the Fréchet Distance between Real-Valued Surfaces. In *28th ACM-SIAM Symp. on Discrete Algo.*, pages 2443–2455. SIAM, 2017.
- [15] J. Scott Carter. Classifying Immersed Curves. *Proceedings of the American Mathematical Society*, 111(1):281–287, 1991.
- [16] Erin Chambers and Yusu Wang. Measuring similarity between curves on 2-manifolds via homotopy area. *29th ACM Symp. Comput. Geom.*, pages 425–434, 2013.
- [17] Erin Chambers, Éric Colin de Verdière, Jeff Erickson, Sylvain Lazard, Francis Lazarus, and Shripad Thite. Homotopic Fréchet distance between curves or, walking your dog in the woods in polynomial time. *J. Comput. Geom.*, 43(3):295–311, 2010.
- [18] Hsien-Chih Chang and Jeff Erickson. Untangling planar curves. *Discrete Comput. Geom.*, 58:889, 2017.
- [19] Hsien-Chih Chang, Brittany Terese Fasy, Bradley McCoy, David L. Millman, and Carola Wenk. From Curves to Words and Back Again: Geometric Computation of Minimum-Area Homotopy. In *Algo. and Data Struct.*, pages 605–619. Springer Nature, 2023.
- [20] Éric Colin De Verdière and Jeff Erickson. Tightening nonsimple paths and cycles on surfaces. *SIAM J. Comput.*, 39(8):3784–3813, 2010.
- [21] Éric Colin De Verdière and Francis Lazarus. Optimal pants decompositions and shortest homotopic cycles on an orientable surface. In *J. of the ACM*, volume 54, page 18, 2007.
- [22] Mark De Berg, Otfried Cheong, Marc Van Kreveld, and Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer Berlin Heidelberg, 2008.
- [23] Hubert de Fraysseix and Patrice Ossona de Mendez. A short proof of a Gauss problem. In *Graph Drawing*, Lecture Notes in Computer Science, pages 230–235, Berlin, Heidelberg, 1997. Springer.
- [24] Max Dehn. Über kombinatorische Topologie. *Acta Mathematica*, 67(0):123–168, 1936.
- [25] Dennis Frisch. Extending immersions into the sphere. 2010. URL <http://arxiv.org/abs/1012.4923>.
- [26] Tamal Dey and Sumanta Guha. Transforming Curves on Surfaces. *J.of Comput. and Syst. Sci.*, 58(2):297–325, 1999.

- [27] Tamal Dey and Haijo Schipper. A new technique to compute polygonal schema for 2-manifolds with application to null-homotopy detection. *Discrete & Comput. Geom.*, 14(1):93–110, 1995.
- [28] Jesse Douglas. Solution of the Problem of Plateau. *Trans. of the American Math. Soc.*, 33(1):263–321, 1931.
- [29] Clifford H. Dowker and Morwen B. Thistlethwaite. Classification of knot projections. *Topology and its Applications*, 16(1):19–31, July 1983.
- [30] Anne Driemel and Sariel Har-Peled. Jaywalking Your Dog: Computing the Fréchet Distance with Shortcuts. *SIAM J. Comput.*, 42(5):1830–1866, 2013.
- [31] Alon Efrat, Leonidas Guibas, Sariel Har-Peled, Joseph Mitchell, and T.M. Murali. New Similarity Measures between Polylines with Applications to Morphing and Polygon Sweeping. *Discrete & Comput. Geom.*, 28(4):535–569, 2002.
- [32] Thomas Eiter and Heikki Mannila. Computing Discrete Frechet Distance. *Tech. Report CD-TR 94/64, Information Systems Department*, 1994.
- [33] David Eppstein. Dynamic generators of topologically embedded graphs. In *15th ACM-SIAM Symp. Discrete Algo.*, pages 599–608, 2003.
- [34] David Eppstein and Elena Mumford. Self-overlapping curves revisited. In *20th ACM-SIAM Symp. Discrete Algo.*, pages 160–169, 2009.
- [35] Jeff Erickson. One-dimensional computational topology lecture notes. Lecture 7, 2020. URL <https://mediaspace.illinois.edu/channel/CS+598+JGE+\0T1\textemdash%C2%A0Fall+2020/177766461/>.
- [36] Parker Evans. On Self-Overlapping Curves, Interior Boundaries, and Minimum Area Homotopies. Bachelor’s thesis, Tulane University, 2018.
- [37] Parker Evans, Brittany Terese Fasy, and Carola Wenk. Combinatorial properties of self-overlapping curves and interior boundaries. *36th Symp. on Comput. Geom.*, 2020.
- [38] Benson Farb and Dan Margalit. *A Primer on Mapping Class Groups*. Princeton University Press, 2011.
- [39] Brittany Terese Fasy, Selçuk Karakoç, and Carola Wenk. On minimum area homotopies of normal curves in the plane. 2017. URL <http://arxiv.org/abs/1707.02251>.
- [40] Efi Fogel, Dan Halperin, and Ron Wein. *CGAL arrangements and their applications: A step-by-step guide*. Springer, 2012.
- [41] Emily Fox, Amir Nayyeri, Jonathan James Perry, and Benjamin Raichel. Fréchet Edit Distance. *40th Symp. Comput. Geom.*, 2024.

- [42] Carl Friedrich Gauss. Nachlass. I. Zur geometria situs. *Werke*, vol. 8, 271–281, 1900.
- [43] Sarel Har-Peled, Amir Nayyeri, Mohammad Salavatipour, and Anastasios Sidiropoulos. How to Walk Your Dog in the Mountains with No Magic Leash. *Discrete & Comput. Geom.*, 55(1):39–73, 2016.
- [44] Strava Inc. www.strava.com, 2022.
- [45] Atlas F. Cook Iv and Carola Wenk. Geodesic Fréchet distance inside a simple polygon. *ACM Trans. on Algos.*, 7(1):1–19, 2010.
- [46] Minghui Jiang, Ying Xu, and Binhai Zhu. Protein Structure–structure Alignment with Discrete Fréchet Distance. *J. Bioinfo. Comput. Bio.*, 06(01):51–64, 2008.
- [47] Selçuk Karakoç. *On Minimum Homotopy Areas*. PhD thesis, Tulane University, 2017.
- [48] Francis Lazarus, Michel Pocchiola, Gert Vegter, and Anne Verroust. Computing a canonical polygonal schema of an orientable triangulated surface. In *17th Symp. Comput. Geom.*, pages 80–89. ACM, 2001.
- [49] Yijing Li and Jernej Barbič. Immersion of self-intersecting solids and surfaces. *ACM Trans. on Graph.*, 45:1–14, 2018.
- [50] Anil Maheshwari and Jiehua Yi. On computing fréchet distance of two paths on a convex polyhedron. *Euro. Work. Comput. Geom.*, pages 41–45, 2005.
- [51] Bruno Martelli. *An introduction to geometric topology*, 2016.
- [52] Morris L. Marx. Extensions of normal immersions of S^1 into R . *Trans. Amer. Math. Soc.*, 187:309–326, 1974.
- [53] Uddipan Mukherjee. Self-overlapping curves: Analysis and applications. *Comput.-Aided Des.*, 40:227–232, 2014.
- [54] James Munkres. *Topology*. Pearson, second edition, 2000.
- [55] Julius v. Sz. Nagy. Über ein topologisches Problem von Gauß. *Mathematische Zeitschrift*, 26(1):579–592, 1927.
- [56] Zipei Nie. On the minimum area of null homotopies of curves traced twice. 2014. URL <http://arxiv.org/abs/1412.0101>.
- [57] Zipei Nie. Private correspondence. 2016.
- [58] Ruth Nussinov and Ann B. Jacobson. Fast algorithm for predicting the secondary structure of single-stranded RNA. *Proc. Natl. Acad. Sci. USA*, 77(11):6309–6313, 1980.
- [59] Valentin Poénaru. Extension des immersions en codimension 1 (d’après Samuel Blank). *Séminaire N. Bourbaki (1966–1968)*, 10:473–505, 1968.

- [60] Tibor Rado. On Plateau's Problem. *Annals of Math.*, 31(3):457–469, 1930.
- [61] Pierre Rosenstiehl and Robert E. Tarjan. Gauss codes, planar hamiltonian graphs, and stack-sortable permutations. *J. of Algos.*, 5(3):375–390, 1984.
- [62] Haijo Schipper. Determining contractibility of curves. In *Eighth ACM Symp. Comput. Geom.*, pages 358–367. ACM Press, 1992.
- [63] Peter Schäfer. Fréchet View - A Tool for Exploring Fréchet Distance Algorithms. *Multimedia Expo. at Symp. Comput. Geom.*, pages 1–5, 2019.
- [64] Peter Shor and Christopher Van Wyk. Detecting and decomposing self-overlapping curves. *Comput. Geom.: Theory and Applications*, 2:31–50, 1992.
- [65] John Stillwell. *Classical Topology and Combinatorial Group Theory*, volume 72 of *Graduate Texts in Mathematics*. Springer, 1993.
- [66] The CGAL Project. *CGAL User and Reference Manual*. CGAL Editorial Board, 5.5.2 edition, 2023. URL <https://doc.cgal.org/5.5.2/Manual/packages.html>.
- [67] Charles J. Titus. The combinatorial topology of analytic functions on the boundary of a disk. *Acta Math.*, pages 106(1–2):45–64, 1961.
- [68] Godfried Toussaint. On separating two simple polygons by a single translation. *Discrete Comput. Geom.*, 4(3):265–278, June 1989.
- [69] Gert Vegter and Chee Yap. Computational complexity of combinatorial surfaces. In *Sixth Symp. Comput. Geom.*, pages 102–111, 1990.
- [70] Carola Wenk. *Shape Matching in Higher Dimensions*. PhD thesis, FU Berlin, 2002.
- [71] Hassler Whitney. On regular closed curves in the plane. *Compos. Math.*, 4:276–284, 1937.