THE UNIVERSITY OF CHICAGO

ALGORITHMS FOR SIMPLE CURVES ON SURFACES, STRING GRAPHS, AND CROSSING NUMBERS

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCES
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

BY
DANIEL ŠTEFANKOVIČ

CHICAGO, ILLINOIS
JUNE 2005

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

We consider algorithmic problems for simple curves on a surface M with non-empty boundary. The surface M is given by a triangulation T such that all vertices of T are on ∂M. The isotopy class of a simple curve is given by normal coordinates - a sequence of intersection numbers of α with the edges of T. This is the *compressed* representation as opposed to the *explicit* representation which lists the intersections and is exponentially larger than the compressed representation

The GEOMETRIC INTERSECTION NUMBER problem is to determine the minimal number of intersections of two curves achievable by deforming the curves. The DEHN-TWIST problem is to compute the Dehn-twist of a simple curve along another closed simple curve. We give the first polynomial-time algorithm for DEHN-TWIST and, as a consequence, for GEOMETRIC INTERSECTION NUMBER, in the compressed representation. The previous algorithms ([Pen84, HTC96]) only allowed computing Dehn-twists along a specific set of simple closed curves.

The STRING GRAPH problem asks if a graph can be realized as an intersection graph of a set of curves in the plane, that is, whether there exists a collection of curves, one for each vertex, such that two curves intersect if and only if the corresponding vertices are adjacent. The problem originated as a layout problem for integrated circuits [Sin66]. The decidability of the STRING GRAPH problem was open for over thirty years. We will first show that the problem is in NEXP, and then, using algorithms for simple curves on surfaces, we will further decrease the complexity to NP, completely resolving the status of the STRING GRAPH problem which turns out to be NP-complete.

The crossing number of a graph G is the minimum number of intersections in a drawing of G in the plane. The odd crossing number of a graph G is the minimum number of pairs of edges that cross odd number of times in a drawing of G. Hanani [Han34] and Tutte [Tut70] showed that if the odd crossing number is zero

then the crossing number is zero as well. Pach and Tóth asked whether the numbers are the same for all graphs. Our study of the intersection numbers has lead us to an infinite sequence of graphs for which the ratio of the odd crossing number and the crossing number approaches $\sqrt{3}/2$. We leave it as an intriguing open question whether ratio smaller than $\sqrt{3}/2$ can be achieved.

ACKNOWLEDGEMENTS

I would like to express my gratitude to all the people who made this thesis possible.

My biggest thanks goes to my parents for bringing me into this world, for their love, and for nurturing my intellectual curiosity.

I was lucky to have Gustáv Nagy as my middle school mathematics teacher. Even after 20 years I still fondly remember some of his weekly puzzles. I wish there were more teachers like him.

My undergraduate advisor Peter Ružička showed me that research in theoretical computer science is fun. He also taught me a lot about encouragement and fairness. I hope that I will be able to pass some of that on my students.

Many thanks Marcus Schaefer for sharing his optimistic approach to research. It has been pleasure to work with Marcus.

My advisor László Babai introduced me to many interesting ideas and research problems, and showed me how to efficiently communicate mathematics. His support during my years in Chicago was invaluable.

Finally, I would like to thank Ivona for all the support and to Žofka for drawing inspiring curves on various surfaces.

CHAPTER 1
INTRODUCTION

1.1 Polynomial-time algorithms for curves on surfaces in compressed representation

Topology is a source of interesting, but often undecidable, algorithmic problems. Some of the decidable problems lead to practical applications; for example, braid groups have been used to construct public key cryptosystems [AAFG01, KLC$^+$00]. The braid group B_n acts naturally on simple curves on the n-times punctured disk, which motivates the study of algorithms on simple curves on surfaces. Another driving force for studying simple curves on surfaces comes from graph drawing - better understanding of simple curves on surfaces may lead to improved algorithms for visualization of graphs.

Algorithmic problems for curves on surfaces have been studied already by Dehn in 1912 (see [Sti99]). Dehn gave algorithms for the Contractibility (deciding if a curve can be continuously contracted to a point) and Transformability problems (deciding if a curve can be continuously transformed to another curve). These problems have been further studied in the CS community, culminating in an optimal algorithm [DG99].

The Geometric Intersection Number problem is to determine the minimal number of intersections of two curves achievable by deforming the curves. Cohen and Lustig [CL87] gave first combinatorial algorithm for computing the geometric intersection numbers on surfaces with non-empty boundary. The algorithm was extended by Lustig [Lus87] to the case of surfaces with no boundary. Another algorithm was given by Hamidi-Tehrani [HT97].

In all the algorithms cited above, the curves are represented either as words in the fundamental group or as sequences of vertices and edges in a triangulation. We will call such representations *explicit*. Simple curves on a surface can be represented more succinctly. Kneser observed in 1930 that the isotopy class of a simple curve in M is determined by the number of intersections with each edge of a triangulation T of M if T has no internal vertices (i. e. all vertices of t lie on ∂M) [Kne30]. The representation is called "normal coordinates." Normal coordinates require space *logarithmic* in the size of the explicit representation. There are other succinct representations of simple curves which are polynomial-time equivalent to normal coordinates: Dehn-Thurston parameterization, "measured train tracks" and compressed representation of words in the fundamental group. We call such representations *compressed*.

A Dehn twist along a simple closed curve γ in a surface is a homeomorphism of the surface to itself obtained by cutting the surface along γ, rotating one of the new boundaries 360 degrees (fixing everything outside a small neighborhood of the boundary) and then gluing the new boundaries back. Penner [Pen84] and Hamidi-Tehrani and Chen [HTC96] gave algorithms to compute Dehn twists along a *particular* set of simple curves. Both algorithms are polynomial-time in the compressed representation. In Section 3.5 we give a polynomial-time algorithm to compute Dehn twist of a simple curve along *any* simple closed curve, where both curves are given in compressed representation. As a consequence we obtain, in Section 3.7, the first polynomial-time algorithm for the GEOMETRIC INTERSECTION NUMBER problem. Previous algorithms used explicit representations and therefore took exponential time [CL87, Lus87, HT97].

The work discussed in Chapter 3 was done jointly with Eric Sedgwick and Marcus Schaefer. Parts appeared in [SSŠ02].

1.2 The string graph problem

The STRING GRAPH problem asks if a graph can be realized as an intersection graph of a set of curves in the plane (i. e., whether there exists a collection of curves, one for each vertex, such that two curves intersect if and only if there is an edge between the corresponding vertices). The problem originated as a layout problem for integrated circuits [Sin66]. Kratochvíl [Kra91] showed that the problem is NP-hard. Kratochvíl and Matoušek gave an example of a graph whose realization requires exponentially many intersections. The decidability of the STRING GRAPH problem was open for over thirty years.

In Section 4.2 we will show that the STRING GRAPH problem is in NEXP. We will study the following problem. Let $\gamma_0, \dots, \gamma_n$ be simple curves where γ_i connects points u_i and v_i. Suppose that $\gamma_1, \dots, \gamma_n$ intersect γ_0 in m points. Is it possible to redraw the curves $\gamma_1, \dots, \gamma_n$ in a regular neighborhood of γ_0 in such a way that (i) γ_i still connects u_i, v_i; (ii) the number of intersections of no pair of curves increases; (iii) the number of intersections on γ_0 strictly decreases. We will show that such a redrawing is possible if $m \geq 2^n$ and the curves are in a plane. This result is tight by an example of Kratochvíl and Matoušek [KM91]. The decidability of the STRING GRAPH problem was also settled, independently, by Pach and Tóth [PT01].

In Section 4.3 we completely resolve the complexity of the STRING GRAPH problem by showing that it is in NP and therefore NP-complete. Handling simple curves given by compressed representation is key to lowering the complexity.

The work discussed in Sections 4.2 and 4.4 was done jointly with Marcus Schaefer. It appeared in [SŠ00, SŠ01, SŠ04]. The work discussed in Section 4.3 was done jointly with Eric Sedgwick and Marcus Schaefer. It appeared in [SSŠ02, SSŠ03].

1.3 Crossing numbers

Crossing number $\mathrm{cr}(G)$ of a graph G is the minimum number of crossings in a drawing of G in the plane (e. g., planar graphs are the graphs with $\mathrm{cr}(G) = 0$). Crossing numbers are used to obtain lower bounds on the chip area in the VLSI design [Lei84]. Computing crossing numbers is difficult even for very regular graphs - for example we do not know the value $\mathrm{cr}(K_{11})$. The main obstacle is that there are no known techniques to prove lower bounds on $\mathrm{cr}(G)$. Since the problem of deciding whether $\mathrm{cr}(G) \leq k$ is NP-hard [GJ83], it is unlikely that general techniques exist.

Odd crossing number $\mathrm{ocr}(G)$ of a graph G is the minimum number of pairs of edges which cross odd number of times in a drawing of G in the plane. Hanani [Han34] and Tutte [Tut70] showed that if odd crossing number of a graph is zero then the graph is planar (i. e., $\mathrm{cr}(G) = 0 \quad \Leftrightarrow \quad \mathrm{ocr}(G) = 0$). Another justification for the concept of odd crossing number is the following: (i) experimental evidence suggests that $\mathrm{ocr}(K_n) = \mathrm{cr}(K_n)$; (ii) The (independent) odd crossing number of a graph can be formulated as a closest codeword problem in a linear code over $\mathbb{Z}/2\mathbb{Z}$. Perhaps the symmetry of K_n can be exploited to solve the closest codeword problem and hence compute $\mathrm{ocr}(K_n)$ (and hopefully $\mathrm{cr}(K_n)$ as well).

Pach and Tóth [PT00b] asked the following natural question: Is $\mathrm{ocr}(G) = \mathrm{cr}(G)$ for every graph G? In Section 5.3 we show an example of a graph G with $\mathrm{cr}(G) \neq \mathrm{ocr}(G)$. In order to construct G we study maps on surfaces. A map R in a surface M is a collection of pairs of points $\{(a_1, b_1), \ldots, (a_n, b_n)\}$ on ∂M such that a_i and b_i lie on different components of ∂M (for $i \in [n]$). We show that for maps on the annulus $\mathrm{cr}(R)$ can be computed by a linear program. This allows us to construct an example of a map R with $\mathrm{ocr}(R)/\mathrm{cr}(R) = \sqrt{3}/2$. This yields an infinite sequence of graphs for which the ratio of the odd crossing number and the crossing number approaches $\sqrt{3}/2$. We leave it as an intriguing open question whether ratio smaller than $\sqrt{3}/2$ can be achieved.

The work discussed in Chapter 5 was done jointly with Michael Pelsmajer and Marcus Schaefer.

CHAPTER 2
PRELIMINARIES

2.1 Model of computation

We will use the **unit-cost word RAM** model (see [Hag98] for a recent survey). The model consists of a finite control where a program is stored and an infinite collection of memory registers $R[0], R[1], \ldots$. A register can hold an integer of absolute value bounded by 2^W. The parameter W is called **word length**. A program is a numbered sequence of instructions. An instruction is one of the following:

- `R[i]:=` $\langle$constant$\rangle$
- `R[i]:=R[j]` op `R[k]` (*arithmetic operation*)
- `R[i]:=R[R[j]]` (*indirect addressing*)
- `goto` $\langle$line number$\rangle$ (*unconditional jump*)
- `if` $R[i] \geq 0$ `then goto` $\langle$line number$\rangle$ (*conditional jump*)

The allowed arithmetic operations are $+, -, \cdot$, and integer division $/$. The operations are assumed to take unit time.

2.2 Words

An **alphabet** Σ is a finite set of symbols. A **word** over alphabet Σ is a finite sequence of elements of Σ. The **length of word** w is denoted $|w|$. Let Σ^k be the set of all words over Σ of length k and let $\Sigma^* = \bigcup_{k\geq 0} \Sigma^k$ be the set of all words over Σ. The **empty word** is denoted ϵ. The set of all words over Σ with the concatenation operation and the empty word form a **monoid** (i. e., a set with associative binary operation and a neutral element). Let $w \in \Sigma^*$ and $1 \leq i \leq j \leq |w|$. The **subword** of w starting at position i and ending at position j is denoted $w[i..j]$.

2.2.1 Straight-line programs

Some words can be described more succinctly than by an explicit sequence of letters - they can be compressed. Perhaps the best known compression methods are Lempel-Ziv-type algorithms [LZ76]. A variant called **LZ1** (see e. g., [FT98]), works as follows. The word w to be compressed is factored as $w = c_1u_1c_2u_2 \dots c_ku_k$, where the $c_i \in \Sigma$ are single letters and u_i is the longest prefix of $u_ic_{i+1} \dots u_k$ appearing (as a substring) in $c_1u_1 \dots c_i$. We encode u_i by giving the position of the first occurrence of u_i in w and the length of u_i. The number k will be called the **size** of the LZ1 encoding of w (the actual number of bits needed to encode w is $\Theta(k(\log|w| + \log|\Sigma|)))$.

The advantage of LZ1 compression is that it is easily computed. The disadvantage is that manipulation of LZ1 encoded strings is relatively complicated, e. g., computing the concatenation of LZ1 compressed strings is non-trivial. Straight-line programs, which we describe next, do not have this shortcoming.

A **straight-line program** is a sequence of assignments to variables $w_1, w_2, \dots, w_n$ (in this order) such that right hand side of the assignment to w_i involves only elements of Σ and variables $w_j,\ j < i$. The word **generated** by a straight-line program is the word w assigned to the last variable (we will also say that w is **given** by the straight-line program). The **size** of a straight-line program is the sum of lengths of the right hand sides of assignments, with the right hand sides are viewed as words over $\Sigma \cup \Omega$, where $\Omega = \{w_1, w_2, \dots, w_n\}$.

Given a word w it is NP-hard to find the smallest straight-line program generating w. In fact the size of the smallest straight-line program is NP-hard to approximate within a factor of 1.0001 ([LS02]). An approximation algorithm with factor $O(\log|w|)$ is known:

Theorem 2.2.1 ([CLL$^+$02, Ryt02]) *Let $w \in \Sigma^*$. Let ℓ be the length of the LZ1 compression of w. Let s be the size of the smallest straight-line program of w. Then*

$$\ell \leq s \leq 2\ell \log|w|.$$

We note that the compression by straight-line programs is very far from universal. We will show that some very natural operations do not preserve compressibility by straight-line programs. Let $u, v \in \{0,1\}^*$ be of the same length. Let $u \vee v \in \{0,1\}^*$ be the bitwise OR of u and v, i. e.,

$$(u \vee v)_i := \begin{cases} 1 & \text{if } u_i = 1 \text{ or } v_i = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Next example shows that $u \vee v$ might need exponentially longer straight-line program than both u and v.

Lemma 2.2.2 *Let p, q be primes, $p < q$. Let $u = (0^{p-1}1)^q$ and $v = (0^{q-1}1)^p$. There are straight-line programs for u and v of size $O(\log pq)$. Let $w = u \vee v$. Any straight-line program for w has size at least $p/2$.*

Proof :

The straight-line programs for u and v are based on repeated squaring (once we computed a word s, computing s^n needs only $\Theta(\log n)$ additional assignments).

Now we show the lower bound on the length of a straight-line program for w.

Claim: Suppose that for $a \neq b$ and k we have $w[a...a+k] = w[b..b+k]$. Then $k \leq 2q - 2$.

Proof: For the sake of contradiction suppose that $k \geq 2q-1$. There exists $c \leq q-1$ such that $q \mid (a+c)$. From the assumption $k \geq 2q-1$ we obtain that $a+c+q \leq a+k$. Note that $w[x] = 1$ iff $x \equiv 0 \pmod p$ or $x \equiv 0 \pmod q$. Hence we have $w[a+c] = w[b+c] = 1$ and $w[a+c+q] = w[b+c+q] = 1$. At most one of $b+c$ and $b+c+q$ can be divisible by p, but each is divisible by either p or q. Hence $q \mid b+c$. We just showed $a \equiv b \pmod q$. A similar argument shows $a \equiv b \pmod p$ and hence $a = b$, a contradiction. □

Thus in the LZ1 encoding $c_1u_1c_2u_2 \dots c_ku_k$ of w each u_i has length at most $2q-1$ and hence $k \geq pq/(2q)$. From Theorem 2.2.1 it now follows that any straight-line program for w has size $\geq p/2$. ■

A straight-line program is **binary** if every right hand side consists of either two variables or a single symbol from Σ. A straight-line program of size s can be transformed to a binary straight-line program of size $\leq 3s$ generating the same word. The transformation replaces assignment $w_i = w_{j_1}...w_{j_k}$ by a sequence of assignments $u_{i,1} = w_{j_1}w_{j_2}$, $u_{i,2} = u_{i,1}w_{j_3}$, $\dots$, $w_i = u_{i,k-2}w_{j_k}$ and can be performed in time linear in s. For the w_{j_i} which are constants from Σ we need to add additional assignment which assigns the constant to a variable.

2.2.2 Algorithms on compressed words

Lemma 2.2.3 *Let w be a word given by a straight-line program P of size n. The following problems can be solved in $O(n)$ time on unit-cost RAM with word size $W = \Omega(\log |w|)$:*

a) *compute the length $|w|$ of w;*

b) *compute the number of occurrences of $a \in \Sigma$ in w;*

c) *given $i, j, 1 \leq i \leq j \leq |w|$, compute a straight-line P' program for $w' = w[i..j]$. The program P' is obtained from P by adding one assignment with right-hand side of length $\leq 6n$.*

Proof :

W. l. o. g., P is binary. The part a) is trivial: if we know lengths of x_a and x_b and the next assignment in P is $x_k = x_a x_b$ then length $|x_k|$ of x_k is $|x_a| + |x_b|$. Similarly we obtain the part b).

Now we show part c). First suppose that $i = 1$, i. e., we want to compute a prefix(x_n, j) (the prefix of $w = x_n$ of length j). If $x_n = x_a x_b$ is the assignment to x_n in P then

$$\text{prefix}(x_n, j) = \begin{cases} \text{prefix}(x_a, j) & \text{if } j \leq |x_a|, \\ x_a \text{prefix}(x_b, j - |x_a|) & \text{otherwise.} \end{cases} \tag{2.1}$$

Using equation (2.1) repeatedly we obtain a word w' over $\{x_1, \dots, x_n\}$ of length $\leq n$ such that prefix(x_n, j) is equal to w'. By the same argument we can get a word for suffix(x_n, j) (the prefix of $w = x_n$ of length j).

Now we want to compute $x_n[i..j]$. If $x_n = x_a x_b$ is the assignment to x_n in P then

$$x_n[i..j] = \begin{cases} x_a[i..j] & \text{if } j \leq |x_a|, \\ x_b[(i - |x_a|)..(j - |x_a|)] & \text{if } i > |x_a|, \\ \text{suffix}(x_a, |x_a| - i + 1)\text{prefix}(x_b, j - |x_a|) & \text{otherwise.} \end{cases} \tag{2.2}$$

Using equation (2.2) repeatedly we reduce the problem to the prefix and suffix cases and obtain a word w' over $\{x_1, \ldots, x_n\}$ of length $\leq 2n$ such that $x_n[i..j]$ is equal to w'. The additional factor of 3 comes from the transformation of P to a binary program. ■

Lemma 2.2.4 *Let w be a word given by a straight-line program P of size n. Let $u_1, \ldots, u_k$ be words with $|u_i| \leq s$ (for all $i \in [k]$). Then in time $O(nks)$ we can compute the number of occurrences of u_i in w (for all $i \in [k]$).*

Proof :

Let $x_1, \ldots, x_n$ be the variables of P. We can compute in time $O(ns)$ the prefix of length s for each x_i and the suffix of length s for each x_i.

For the x_i with $|x_i| \leq 2s$ we compute the number of occurrences of the u_i explicitly (e. g., using the KMP algorithm). Now let $x_\ell := x_j x_k$ be an assignment in P, where $|x_\ell| > 2s$. Suppose that we already know the number of occurrences of the u_i in x_j and x_k. To compute the number of occurrences of the u_i in x_ℓ we only need to count the occurrences of the u_i which intersect the boundary between x_j and x_k. Again we use the KMP algorithm on the string suffix(x_j, s)prefix(x_k, s). ■

Theorem 2.2.5 ([MST97]) *Let u, v be words given by straight-line programs P_u and P_v. Let m be the size of P_u and n be the size of P_v. There is a deterministic algorithm which decides whether $u = v$ in time $O(m^2n^2)$ on unit-cost word RAM with word size $W = O(m + n)$.*

Theorem 2.2.6 ([GKPR96]) *Let u, v be words given by straight-line programs P_u and P_v. Let m be the size of P_u and n be the size of P_v. There is a randomized algorithm which*

- *if* $u = v$ *always answers* YES,
- *if* $u \neq v$ *answers* NO *with probability* $\geq 1 - \delta$.

The algorithm runs in time $O(m + n)$ *on unit-cost word RAM with word size* $W = O(\log 1/\delta)$.

2.2.3 Word equations

Let Σ be an alphabet of symbols and Ω be an alphabet of variables. A **system of equations** E is a set of pairs (u_L, u_R) of words $u_L, u_R \in (\Sigma \cup \Omega)^*$. The **size** $|E|$ of the system of equations is the sum of $|u_L| + |u_R|$ summed over $(u_L, u_R) \in E$. A system of equations is **quadratic** if each variable $x \in \Omega$ occurs at most twice in the system.

A **morphism** h is a function $h : \Sigma^* \to \Sigma'^*$ such that $h(uv) = h(u)h(v)$ for all $u, v \in \Sigma^*$. A **solution** of a system of equations E is a morphism $h : (\Sigma \cup \Omega)^* \to \Sigma^*$ such that h is identity on Σ^* and $h(u_L) = h(u_R)$ for each $(u_L, u_R) \in E$. The **size** of the solution is $\sum_{x \in \Omega} |h(x)|$.

Word equations were shown to be decidable by Makanin [Mak77]. The upper bound on the complexity of word equations was improved by Plandowski:

Theorem 2.2.7 ([Pla99]) *The following problem can be solved in* PSPACE*:*
INSTANCE: E *a system of word equations.*
QUESTION: Does E *have a solution?*

The problem is NP-hard even for quadratic word equations (see e. g., [RD99]). Plandowski and Rytter [PR98] showed that if the following (plausible) conjecture is true then word equations are in NP (and hence NP-complete).

Conjecture 1 ([PR98]) *There exists a constant* C *such that if a system of equations* E *of size* n *has a solution then it has a solution of size* C^n*.*

Conjecture 1 is open even in the case of quadratic word equations.

A **system of equations with length constrains** is a system of equations together with a function $\ell : \Omega \to \mathbb{N}$. The solution has to respect the lengths, i. e., we require $|h(x)| = \ell(x)$.

Let Γ be a set of symbols disjoint from Ω and Σ. The **most general solution** of a system of equations E with length constrains ℓ is a solution g over $\Sigma \cup \Gamma$ (i. e., $g(x) \in (\Sigma \cup \Gamma)^*$) such that for every solution h over Σ^* there exists $f : \Gamma \to \Sigma$ such that $h(x) = f(g(x))$, $x \in \Omega$.

Once we have the lengths, finding the solutions is easy:

Theorem 2.2.8 ([PR98]) *Let E be a system of word equations with length constrains ℓ. The most general solution of E can be found in time polynomial in $|E| + \sum_{x\in\Omega} \ln \ell(x)$.*

In the next section we will see that for quadratic systems of word equations we have a linear-time algorithm.

2.2.4 Diekert-Robson algorithm

Diekert and Robson [RD99] gave an algorithm which finds a straight-line program P for the most general solution of a system of quadratic word equations with length constrains $\ell : \Omega \to \mathbb{N}$.

Theorem 2.2.9 ([RD99]) *Let E be a quadratic system of word equations with length constrains ℓ. The most general solution of E can be found in time $O(|E| + \sum_{x\in\Omega} \ln \ell(x))$ on a unit-cost word RAM with word size $W = \Omega(\ln \sum_{x\in\Omega} \ell(x))$.*

The algorithm was extended to handle involutions by Diekert and Kufleitner [DK02]. We include a short description of the algorithm.

An **involution** is a function $\Sigma \to \Sigma$, $a \mapsto \overline{a}$ such that $\overline{\overline{a}} = a$ for all $a \in \Sigma$. Let $\overline{\Omega}$ be a disjoint copy of Ω. We define a (formal) involution on $\Omega \cup \overline{\Omega}$ by $x \mapsto \overline{x}$ and $\overline{x} \mapsto x$ for all $x \in \Omega$. The formal involution is extended to $(\Sigma \cup \Omega \cup \overline{\Omega})^*$ by

$u_1 u_2 \ldots u_k \mapsto \overline{u}_k \ldots \overline{u}_2 \overline{u}_1$. We will only consider two special cases of involutions: **reversal** (which fixes all $a \in \Sigma$) and **inverse** (which fixes no $a \in \Sigma$).

Theorem 2.2.10 ([DK02]) *Let E be a quadratic system of word equations with length constrains ℓ. The most general solution of E can be found in time $O(|E| + \sum_{x \in \Omega} \ln \ell(x))$ on a unit-cost word RAM with word size $W = \Omega(\ln \sum_{x \in \Omega} \ell(x))$.*

Let E be a system of equations with involution. W. l. o. g., we can assume that all the equations are of the form $z = xy$, $z \in \Omega$, $x, y \in \Omega \cup \overline{\Omega} \cup \Sigma$, $\ell(x), \ell(y) \geq 1$ and $\ell(z) = \ell(x) + \ell(y)$. If $z \in \Omega$ occurs twice on the left hand side of an equation, we say that z is **doubly defined**.

If there are no doubly defined variables then the equations are the straight-line program for the most general solution (all the undefined variables can be set arbitrarily) and we are done. Suppose that some z is doubly defined. The algorithm builds straight-line program P for the most general solution by executing the following steps. If a variable is removed from E then its definition is added to P.

- Balancing: Equations $\{z = xy, x = pq\}$ with $\ell(qy) \leq 0.45 \cdot \ell(z)$ are replaced by $\{z = pr, r = qy\}$ (where r is a new variable, $\ell(r) = \ell(qy)$). The balancing step is applied in the mirror symmetric case as well (i. e., to equations $\{z = xy, y = pq\}$ with $\ell(xp) \leq 0.45 \cdot \ell(z)$).

- Reduction: This step is applied only if the balancing step cannot be applied. We have $z = xy$ and $z = uv$. W. l. o. g., $\ell(x) \geq \ell(u)$.

  - If $\ell(x) = \ell(u)$ then occurrences of $x, \overline{x}$ are replaced by $u, \overline{u}$ and occurrences of $y, \overline{y}$ are replaced by $v, \overline{v}$.

  - If $\ell(x) > \ell(u)$ then equations $\{z = xy, z = uv\}$ are replaced by $\{x = uw, v = wy\}$ (where w is a new variable, $\ell(w) = \ell(x) - \ell(u)$). If moreover

    * $x = v$ and $\ell(w) \geq \ell(z)/2$ then $\{x = uw, v = wy\}$ are replaced by $\{u = rs, y = sr\}$ (where r, s are new variables with $\ell(r) = \ell(x) \pmod{\ell(u)}$, $\ell(s) = \ell(u) - \ell(r)$) and

* $x = \overline{v}$ then $\{x = uv, v = wy\}$ are replaced by $\{w = sz\overline{s}\}$ (where s, z are new variables with $\ell(z) = \ell(w) \pmod 2$, $\ell(s) = \lfloor \ell(w)/2 \rfloor$).

2.2.5 Extensions

Let Σ be an alphabet. Let G be a graph on Σ. We assume that if a and b are connected then a and $\overline{b}$ are connected as well. A **trace monoid** given by G is defined by $\Sigma^*/\{ab = ba \mid \{a, b\} \in G\}$, (i. e., the adjacent vertices in Σ can commute). Matiyasevich showed in 1996 that the solvability of equations in trace monoids is decidable [DMM97]. We will need the following result which allows involutions in the equations.

Theorem 2.2.11 ([DK02]) *The following problem can be solved in* PSPACE*:*
INSTANCE: E be a quadratic system of word equations with involution, alphabet Σ, graph G on Σ.
QUESTION: Does E have a solution over the trace monoid given by G?

2.3 Background on topology

In this section we review the definitions and results of topology that are used in the dissertation. We refer the interested reader to [Sti93, Mas91, Lee00, FM97] for more details.

2.3.1 General topological spaces

Topological space, continuous map, connectivity

A **topological space** is a pair (X, O_X), where O_X is a set of subsets of X such that $0, X \in O_X$ and O_X is closed under the operations of union and finite intersection. The elements of O_X are called **open sets**. A topological space X is called **Hausdorff** if for every pair of distinct points $p, q \in X$ there exist disjoint open sets $P, Q \in O_X$ such that $p \in P$ and $q \in Q$. A collection of open sets B is called **basis** of X if every open set can be written as union of sets from B. A topological space is **compact** if every covering by open sets has a finite subcovering. In the following we will only consider topological spaces which are Hausdorff, compact and have countable basis.

A function $f : X \to Y$ between two topological spaces spaces is **continuous** if the preimage of every open set is open (i. e., $f^{-1}(O_Y) \subseteq O_X$). A continuous function is called a **map**. A **homeomorphism** of topological spaces X and Y is a bijection $f : X \to Y$ such that both f and f^{-1} are continuous.

A topological space X is **connected** if there are no disjoint open sets $P, Q \in O_X$ such that $X = P \cup Q$. A **path** α is a map of $\alpha : [0, 1] \to X$. A topological space X is **path-connected** if for any $p, q \in X$ there is a path from p to q.

Homotopy, embedding, isotopy

Two maps $f, g, : X \to Y$ are **homotopic** it there is a map $H : X \times [0, 1] \to Y$, called **homotopy**, such that $H(\cdot, 0) = f(\cdot)$ and $H(\cdot, 1) = g(\cdot)$. We say that a homotopy H **fixes** $p \in X$ if $H(p, \cdot)$ is a constant function.

Two spaces X, Y are **homotopy equivalent** if there are maps $f : X \to Y$ and $g : Y \to X$, called **homotopy equivalences**, such that $f \circ g$ is homotopic to the identity map $i_X : X \to X$ and $g \circ f$ is homotopic to the identity map $i_Y : Y \to Y$. A space homotopy equivalent to a single point is called **contractible**.

An injective map $f : X \to Y$ is called an **embedding**. A homotopy $H : X \times [0, 1] \to Y$ is called an **isotopy** if $h_t(\cdot) = H(\cdot, t)$ is an embedding for every $t \in [0, 1]$. Two embeddings $f_1, f_2 : X \to Y$ are **ambient isotopic** if there is a continuous deformation of Y (the ambient space) moving f_1 to f_2 (i. e., if there is a map $H : Y \times [0, 1] \to Y$ such that $H(\cdot, 0)$ is the identity, $H(\cdot, t) : Y \to Y$ is a homeomorphism and $H(\cdot, 1) \circ f_1 = f_2$).

Sometimes it is convenient to specify a set P of points in Y, called **punctures**. If there are punctures in Y we require that a homotopy H does not "pass through" the punctures, i. e., $H : X \times [0, 1] \to Y \setminus P$.

Manifolds

A d-**manifold** M (with boundary) is a Hausdorff topological space with a countable basis of open sets where each point has a neighborhood homeomorphic to an open disk in $\mathbb{R}^d$ or $R^d_+ = \{(x_1, \ldots, x_d) \in \mathbb{R}^d \mid x_d \geq 0\}$. We will only consider compact manifolds. The **boundary** ∂M of M is the set of points which do not have neighborhood homeomorphic to $\mathbb{R}^d$. If $\partial M = \emptyset$ we say that M is **closed**.

Given a topological space X and an equivalence relation $\sim$ on X the **quotient space** $X/\sim$ is defined as follows: the points of $X/\sim$ are the equivalence classes $\{[x] \mid x \in X\}$ of $\sim$ in X and $U \subseteq X/\sim$ is open iff $f^{-1}(U)$ is open (where $f : x \mapsto [x]$). Let M_1, M_2 be disjoint d-manifolds. Let D_1, D_2 be d-disks embedded in M_1, M_2 and let $h : D_1 \to D_2$ be a homeomorphism. The **connected sum** $M_1 \# M_2$ is the quotient space of $M_1 \cup M_2$ with $x \sim h(x)$. The connected sum is independent (up to homeomorphism) on the choice of D_1, D_2 and h.

Simplicial complexes

A n-**simplex** Δ^n is a convex hull of a set of $n+1$ points $\{p_1, \ldots, p_{n+1}\}$ in general position in some $\mathbb{R}^m$, $m \geq n$. A subset of $\ell+1$ points from $\{p_1, \ldots, p_{n+1}\}$ determines a ℓ-**dimensional face**. An n-simplex is a n-manifold. The boundary $\partial\Delta^n$ is the union of $n-1$-dimensional faces of Δ^n. A n-**simplicial complex** C is a union of simplices of dimension $\leq n$ such that any two simplices are either disjoint or they intersect in a common face.

An **orientation** of a n-simplex Δ^n is a ordering of the vertices $\{p_1, \ldots, p_{n+1}\}$ with two orderings being equivalent if they differ by an even permutation. An **orientation** of a simplicial complex C is an ordering for each simplex in C such that if two n-simplices share an $n-1$-face then the induced orientations on the $n-1$ face are opposite. A complex is **orientable** if there exists an orientation of the complex. Orientability is a topological invariant (i. e. if C_1 and C_2 are homeomorphic then either both are orientable or both are non-orientable).

Definition 2.3.1 A **triangulation** of a topological space X is a simplicial complex T together with a homeomorphism $h : T \to C$.

Theorem 2.3.2 ([Rad25]) *Every 2-manifold can be triangulated.*

Theorem 2.3.3 ([Moi52]) *Every 3-manifold can be triangulated.*

Not every 4-manifold can be triangulated (see [AM90]).

Let C be a n-simplicial complex. Let $d_i,\ i = 0, \ldots, n$ be the number of i-simplices in C. The **Euler characteristic** of C is

$$\chi(C) = \sum_{i=0}^{n} (-1)^i d_i.$$

The Euler characteristic is a topological invariant.

2.3.2 Surfaces

Paths, loops, fundamental group

Let M be a surface. Recall that a **path** α is a map of $\alpha : [0, 1] \to M$. If α and β are paths such that $\alpha(1) = \beta(0)$ then their **product** is the path $\gamma : [0, 1] \to M$ defined by $\gamma(x) := \alpha(2x)$ and $\gamma(x + 1/2) := \beta(2x)$ for $x \in [0, 1/2]$. The **inverse** of a path α is the path α^{-1} defined by $\alpha^{-1}(x) := \alpha(1 - x)$ for $x \in [0, 1]$.

Let α, β be paths with the same start point $\alpha(0) = \beta(0)$ and the same end point $\alpha(1) = \beta(1)$. We say that paths α and β are **homotopic** if there is a homotopy between α and β, fixing points 0 and 1. The relation of homotopy equivalence (of paths) is an equivalence relation, the equivalence classes are called **homotopy classes of paths**.

A **loop** is a path such that $\alpha(0) = \alpha(1)$. We say that $\alpha(0)$ is the **basepoint** of the loop α. Let $p \in M$ be a point on M. Consider the loops with basepoint p. The set of homotopy classes of such loops, together with the product operation and inverse operation, is a group, called the **fundamental group** $\pi_1(M, p)$ of M. Up to isomorphism the group does not depend on the choice of the basepoint p. A loop α is

null-homotopic if it is homotopic to a point (by a point we mean the constant loop β, $\beta(\cdot) := p$). The homotopy class of null-homotopic loops is the identity element in $\pi_1(M, p)$.

Curves

Let $S^1 = [0,1]/\{0 \sim 1\}$. A **closed curve** is a map $\alpha : S^1 \to M$ such that

$$\operatorname{Im} \alpha \cap \partial M = \emptyset.$$

Unlike a loop a closed curve has no distinguished point. If $\alpha(0) = p$ then α can be viewed as an element of $\pi_1(M, p)$. The notions of homotopy for closed curves and for loops are different. If the notion used is not clear from the context we will call the homotopy of loops **homotopy with fixed basepoint**, and the homotopy of curves **free homotopy**.

Lemma 2.3.4 *Let α, β be closed curves containing a point p. Then α and β are (freely) homotopic iff α and β are conjugate in $\pi_1(M, p)$.*

A closed curve is **simple** if the map α is an embedding (i. e., it is injective). A closed curve is **boundary-homotopic** if it is homotopic to a closed curve contained in ∂M. A closed curve is **trivial** if it is null-homotopic or boundary-homotopic.

An **arc** is a path $\alpha : [0,1] \to M$ such that $\operatorname{Im} \alpha \cap \partial M = \{\alpha(0), \alpha(1)\}$. An arc is a **simple** if the map α is an embedding (i. e., it is injective). An arc is **trivial** if it is homotopic to a path contained in ∂M. Two arcs α, β are **homotopic rel boundary** if there is a homotopy which does not move the points on the boundary (i. e., $\alpha(0), \alpha(1), \beta(0), \beta(1)$).

A **curve** is either an arc or a closed curve. A curve which is not trivial is called **essential**. A **simple curve** is either a simple arc or a simple closed curve. A **simple multi-curve** is a collection of disjointly embedded simple curves. Let $\mathrm{CS}(M)$ be the set of all isotopy classes of simple multi-curves on M and let $\mathrm{C}S_0(M) \subseteq \mathrm{C}S(M)$ be the set of isotopy classes of simple multi-curves whose components are simple closed curves.

Classification of surfaces

A **surface** is a 2-manifold. The surfaces from which all other surfaces can be "built" are the **torus** ($[0,1]\times[0,1]$ quotiented by $(0,y)\sim(1,y)$ and $(x,0)\sim(x,1)$, $x,y\in[0,1]$) and the **projective plane** ($[0,1]\times[0,1]$ quotiented by $(0,y)\sim(1,1-y)$ and $(x,0)\sim(1-x,1)$, $x,y\in[0,1]$). Let M_g (the closed orientable surface of **genus** g) be the connected sum of g tori. Let M'_g (the closed non-orientable surface of **genus** g) be the connected sum of g projective planes). Let $M_{g,b}$ (resp. $M'_{g,b}$) be obtained from M_g (resp. M'_g) by removing interiors of b disjoint disks.

Theorem 2.3.5 (Classification of closed surfaces) *Each closed surface is homeomorphic to exactly one of the $M_{g,b}$ for some $g\geq 0, b\geq 0$, or one of the $M'_{g,b}$ for some $g\geq 1, b\geq 0$.*

We have $\chi(M_{g,b})=2-2g-b$ and $\chi(M'_{g,b})=2-g-b$. A surface M is called **hyperbolic** if $\chi(M)<0$. The **pair of pants** $M_{0,3}$ is the simplest hyperbolic surface.

Jordan-Schönflies Theorem

Theorem 2.3.6 (Jordan-Schönflies Theorem) *Let γ be a simple closed curve in $\mathbb{R}^2$. There is a homeomorphism of $\mathbb{R}^2$ to $\mathbb{R}^2$ mapping γ to the unit circle.*

A simple closed curve $\alpha\subseteq M$ is called **separating** if $M-\alpha$ has two disjoint components. Note that by the Jordan-Schönflies theorem every simple closed curve on a sphere (or a sphere with holes) is separating.

Intersection Numbers

Let M be an orientable surface. Let α_1,α_2 be two simple curves on M. The **geometric intersection number** of α_1,α_2 is

$$i(\alpha_1,\alpha_2)=\min_{\gamma_i\in[\alpha_i]}|\gamma_1\cap\gamma_2|,$$

where $[\alpha_i]$ is the set of curves isotopic (rel boundary) to α_i. The curves α_1, α_2 are called **isotopy disjoint** if $i(\alpha_1, \alpha_2) = 0$.

Let γ_1, γ_2 be simple curves. A **bigon** B bounded by γ_1, γ_2 is a disc which has exactly two intersections (of γ_1 and γ_2) on the boundary ∂B, see Figure 2.1.

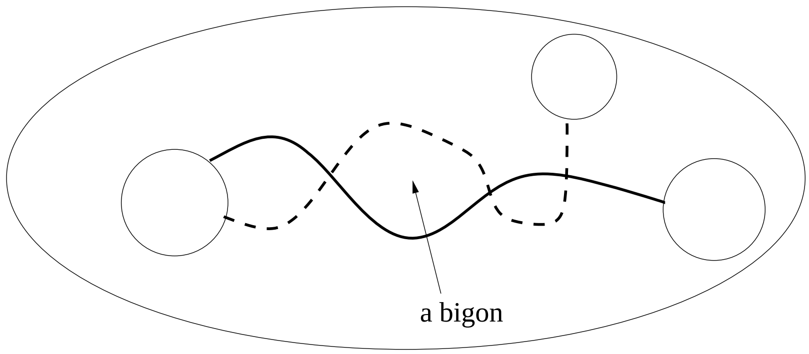

Figure 2.1: Example of a bigon.

We will need the following standard result (see e.g. [FT91]). The condition that the manifold M is orientable is necessary.

Lemma 2.3.7 *Let M be an orientable surface. If two simple curves γ_1, γ_2 on M intersect more than $i(\gamma_1, \gamma_2)$ times then they bound a bigon.*

Lemma 2.3.8 *Let M be an orientable surface. Let $\alpha_1, \dots, \alpha_n$ be simple curves on M. There exist simple curves $\gamma_1, \dots, \gamma_n$ on M such that $\gamma_i \sim \alpha_i,\ i \in [n]$ and*

$$|\gamma_i \cap \gamma_j| = i(\alpha_i, \alpha_j),\ \ i, j \in [n].$$

Proof :

Let $\gamma_1, \dots, \gamma_n$ be in general position such that $\sum |\gamma_i \cap \gamma_j|$ is minimized. If there are two properly embedded arcs γ_i, γ_j which intersect more than $i(\alpha_i, \alpha_j)$ times then, by Lemma 2.3.7, they bound a bigon B. Let e, f be the sides of bigon B. W.l.o.g., assume that B is the smallest bigon w.r.t. containment. Then any properly embedded arc c which crosses e also crosses f, otherwise B would not be smallest. However

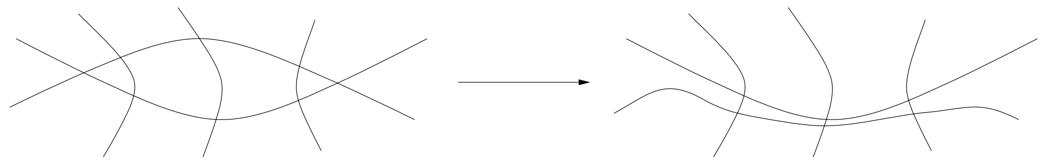

Figure 2.2: Decreasing the number of intersections if a pair of curves bounds a bigon.

then we can reroute the properly embedded arc γ_i to decrease the total number of intersections, a contradiction. See Figure 2.2. ■

Let M be an orientable surface. Let α_1, α_2 be two simple curves on M. Suppose that both α_1, α_2 are orientated. We assign $+1$ to crossings of α_1, α_2 in which α_2 passes from left to right and -1 to the crossings in which α_2 passes from right to left. The **algebraic intersection number** $\hat{i}(\alpha_1, \alpha_2)$ of α_1, α_2 is the sum of the assigned values. The number depends only on the isotopy classes of α_1, α_2 (see e. g., [Ful95]). We have $i(\alpha_1, \alpha_2) \equiv \hat{i}(\alpha_1, \alpha_2) \pmod 2$.

Dehn twists

Let α be a simple closed curve in M. A **Dehn twist** along α is a homeomorphism $D_\alpha : M \to M$ obtained by cutting M along α, rotating one of the copies of α, 360 degrees and gluing them back. More precisely, if an annulus around α is parameterized by $\{(x, \varphi), 1 \le x \le 2, 0 \le \varphi \le 2\pi\}$ then D_α is $(x, \varphi) \mapsto (x, \varphi + 2\pi(x-1) \pmod{2\pi})$ on the annulus and identity elsewhere.

Dehn twists are interesting because they generate the **mapping class group** $\mathcal{M}_g^s$ (the group of orientation preserving homeomorphisms of the surface M_g^s to itself modulo isotopy). Dehn showed that $\mathcal{M}_g$ is generated by finitely many Dehn twists. Lickorish [Lic64, Lic66] gave a set of $3g-1$ generators of $\mathcal{M}_g$.

Polygonal δ-skeleton

A **region** is a subset of the plane homeomorphic to the closed unit disc. Note that both a region and its boundary are compact (the homeomorphic image of a compact set is compact). The boundary of a region is a simple closed curve.

The **Hausdorff distance** $\mathrm{dist}(A, B)$ of two sets is defined as

$$\mathrm{dist}(A, B) = \max\{\sup_{x \in A} \inf_{y \in B} d(x, y), \sup_{y \in B} \inf_{x \in A} d(x, y)\},$$

where $d(x, y)$ is the Euclidean distance of two points in the plane. The Hausdorff distance is a metric for compact sets, i.e. it is symmetric, satisfies the triangle inequality and $\mathrm{dist}(A, B) = 0$ iff $A = B$. We let

$$d(A, B) = \inf_{x \in A} \inf_{y \in B} d(x, y).$$

Note that for closed, nonempty sets $d(A, B) > 0$ iff $A \cap B = \emptyset$. For sets d is not a metric.

A **polygonal arc** is a curve formed by a finite number of line segments (a polygonal arc can self-intersect).

Consider a simple curve $\gamma : [0, 1] \to \mathbb{R}^2$. Let $C = \mathrm{Im}\, \gamma$. A **polygonal δ-skeleton** for C is a polygonal arc connecting consecutive points of $(\gamma(r_i))_{1 \le i \le n}$, where $0 = r_1 < \ldots < r_n = 1$ are such that $r_{i+1} - r_i < \delta$. The points $\gamma(r_i)$ are the vertices of the polygonal skeleton.

The homeomorphism $\gamma : [0, 1] \mapsto C$ is uniformly continuous (being defined on a compact set) which immediately implies the following result (for a proof see [Tri95]).

Proposition 2.3.9 *Given a simple curve C and $\varepsilon > 0$ there is $\delta > 0$ such that every polygonal δ-skeleton P of C satisfies* $\mathrm{dist}(P, C) < \varepsilon$.

2.4 Representing surfaces

In order to define an algorithmic problem it is necessary to have a combinatorial description of the input. In this section we review common combinatorial representations of surfaces (triangulation, pants decomposition, and polygonal schema), and ways of representing isotopy classes of simple curves in surfaces (normal coordinates, intersection sequence, a word in the fundamental group, Dehn-Thurston coordinates, and π_1-train-tracks).

2.4.1 Triangulation

A **graph** G is a pair of sets (V_G, E_G), where E is a set of 2-element subsets of V. The elements of V are called **vertices** and the elements of E are called **edges**. A **multi-graph** G is a set V and a multi-set E, where E contains 2-element and 1-element subsets of V.

A **triangulation** of a surface M is an embedding of a graph $T = (V_T, E_T)$ into M such that each face f is a triangle (i. e. a disk such that in a closed walk along the boundary of f we meet 3 points from V_T). A **pseudo-triangulation** is an embedding of a multi-graph G into M such that each face is a triangle. Note that in pseudo-triangulation the points encountered by the walk along the boundary do not have to be distinct. The vertices of V_T which are on ∂M are called a **boundary vertices**, the other vertices are called **interior vertices**. Similarly $e \in E_T$ is **boundary edge** if it is contained in ∂M and **interior edge** otherwise.

2.4.2 Intersection sequence

Let M be a surface and let T be a pseudo-triangulation of M. For each edge $e \in T$ we fix an orientation. The edge e with opposite orientation will be referred to as e^{-1}. Let α be an oriented simple arc on M in general position w.r.t. T. We assign a word $w_\alpha \in (E_T \cup E_T^{-1})^*$ to α as follows. Traverse α and for each crossing of an edge $e \in E_T$ append e^x to w_α, where x is the sign of the crossing. Note that exactly two edges in w_α are boundary. The word w_α is called the **intersection sequence** of α with T.

For a simple closed curve α on M the intersection sequence w_α is a cyclic word defined analogously. All edges occurring in w_α are interior.

Note that e and e^{-1} are adjacent in w_α then α and e form a bigon B. Removal of B (by pulling α) corresponds to cancellation of the adjacent e and e^{-1}. A simple curve is called **normal** w.r.t. T if w_α is reduced (cyclically reduced in the case of closed curve).

Observation 2.4.1 *Let M be a surface and let T be a pseudo-triangulation of M. For every simple curve α in M there exists γ isotopic to α such that γ is normal w.r.t. T.*

Note that if T has no inner vertices then in any isotopy $\alpha_t,\ t \in [0,1]$ (in general position w.r.t T) the reduction of words w_{α_t} is the same for all $t \in [0,1]$ (except for finitely many t for which α_t is not in general position and hence w_{α_t} is not defined).

2.4.3 Normal coordinates

Given a simple curve α in normal position w.r.t. T let $\alpha(e)$ be the number of intersections of α and $e \in T$. The vector $(\alpha(e))_{e \in T}$ is called **normal coordinates** of α. If two simple curves α, β in normal position w.r.t. T have the same normal coordinates then they are isotopic. If the triangulation T has no inner vertices isotopic simple curves normal w.r.t. T have the same normal coordinates.

The complexity of the representation of α using normal coordinates is the total bitlength of the $\alpha(e)$, $e \in T$. Note that the size is exponentially smaller than the length of the intersection sequence.

2.4.4 Pants decomposition and Dehn-Thurston coordinates

A **pair of pants** $P = M_{0,3}$ is a sphere with three holes. A **pants decomposition** of a surface $M = M_{g,h}$ is a simple multi-curve C such that the connected components of $M \ltimes C$ (M cut along C) are pairs of pants. A simple Euler characteristic computation shows that C has $3g-3+h$ simple curves and $M \ltimes C$ consists of $2g-2+h = -\chi(M)$ pairs of pants. Every surface of negative Euler characteristic has a pants decomposition. If a surface is not a pair of pants and it is of negative Euler characteristic then it has infinitely many different isotopy classes of pants decomposition. Any two isotopy classes of pants decompositions are joined by a sequence of moves in which only one curve in C changes at a time [HT80].

A closed curve in a pair of pants P is either null-homotopic or homotopic to the boundary. Thus a simple multi-curve in P consists only of simple arcs. There are 6 isotopy (with free boundary) classes of non-trivial properly embedded arcs on P, one for each pair of (not necessarily distinct) holes.

An isotopy class of a simple multi-curve α in P is determined by three non-negative integers a_1, a_2, a_3, the numbers of points of α on the boundary components of P. There is a simple multi-curve in P for any $a_1, a_2, a_3 \geq 0$ with $a_1 + a_2 + a_3$ even.

Consider the triangulation T of P from Figure 2.3. Assign weight 1 to "thin" edges of T and $1 + \varepsilon$ to "thick" edges, where ε is infinitesimally small. In each isotopy class of a simple multi-curve on P there is a unique representative (up to normal isotopy w.r.t. T) of minimal weighted intersection with T. The uniqueness of the representative follows from the uniqueness of the representative for the 6 isotopy classes of properly embedded arcs in P.

For a simple multi-curve α in P given by a_1, a_2, a_3, the normal coordinates in T of the representative of α can be easily computed in constant time; this is done by taking the appropriate linear combination of the normal coordinates of the representatives of the 6 isotopy classes of properly embedded arcs in P.

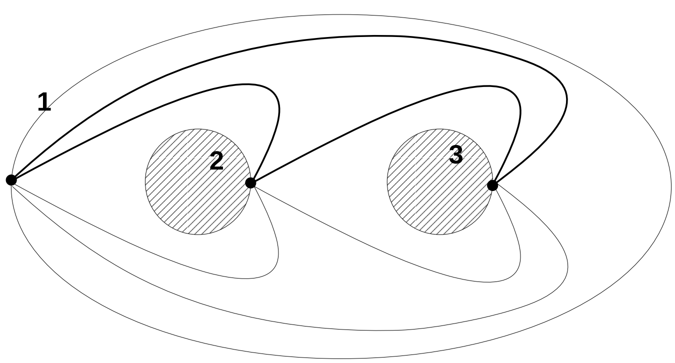

Figure 2.3: Triangulation T of a pair of pants P.

Dehn-Thurston coordinates parameterize the set of curves on M_g^s by a subset of $\mathbb{Z}^m$ where $m = 6g - 6 + 2s$. We have a pair of pants decomposition C of M and on each curve of C we have an interval called a window. Given a simple multi-curve α, the first $3g - 3 + h$ coordinates are the geometric intersection numbers of α with the components of C. The remaining $3g-3+h$ describe how α "twists" within a small neighborhood of each of the component of C (after the curve is isotoped to a certain normal form in each pair of pants).

2.4.5 Polygonal schema and π_1-train tracks

A convenient way of representing a surface, called a **polygonal schema**, is as follows. Take a polygon with an even number of edges, pair the edges and glue the paired edges together. There are two different ways to glue a pair of edges depending on their relative orientation. The construction can be encoded by a word w. Let Σ be an alphabet with n symbols. Let w be a word over $\Sigma \cup \Sigma^{-1}$ of length $2n$ such that for each $a \in \Sigma$ there are either two occurrences of a in w or one occurrence of both a and a^{-1}. The word w defines a pairing (a is paired with the other a or a^{-1}) as well as an orientation of edges of a polygon P with $2n$ edges.

Let P be a polygonal schema and M the surface it represents. We will often regard the images of vertices of P as punctures in M. Note that many vertices of P can map to the same puncture. Every surface with at least one puncture can be represented by a polygonal schema. The **canonical polygonal schema** of an orientable surface of genus g (with $p \geq 1$ punctures) is given by the word

$$a_1 b_1 a_1^{-1} b_1^{-1} \dots a_g b_g a_g^{-1} b_g^{-1} c_1 c_1^{-1} \dots c_{p-1} c_{p-1}^{-1}.$$

The **canonical polygonal schema** of a non-orientable surface of genus g (with $p \geq 1$ punctures) is given by the word

$$a_1 a_1 \dots a_g a_g c_1 c_1^{-1} \dots c_{p-1} c_{p-1}^{-1}.$$

We will consider only orientable surfaces.

Pick a base point p in the interior of M. The fundamental group of the punctured surface M is the free group generated by curves α_a, $a \in \Sigma$, the curve α_a crosses only the edge a of P.

An (isotopy class of) simple multi-curve α on M can be given by a π_1**-train track**, which is essentially a symmetric $2n \times 2n$ matrix A, where A_{ij} is the number of strands going from edge i to edge j (i, j are edges of P). Since α is simple we have

$$A_{ik}A_{j\ell} = 0, \text{ for } 1 \leq i < j < k < \ell \leq 2n. \tag{2.3}$$

If both i and j are labeled by symbols a or a^{-1} then the number of strands entering i must be the same as the number of strands entering j, i. e.,

$$\sum_{k=1}^{2n} A_{ik} = \sum_{k=1}^{2n} A_{jk}. \tag{2.4}$$

There is a one-to-one correspondence between homotopy classes of simple multi-curves on M and non-negative, integral, symmetric matrices satisfying conditions (2.3) and (2.4).

2.4.6 Fundamental group

The fundamental group of an orientable surface $M_{g,h}$ with non-empty boundary ($h > 0$) is a free group with $2g + h$ generators. The fundamental group of orientable boundaryless surface M_g has presentation

$$\langle a_1, b_1, \ldots, a_g, b_g \mid [a_1, b_1] \ldots [a_g, b_g] = 1 \rangle,$$

where $[a, b] = aba^{-1}b^{-1}$ is the **commutator** of a and b. The fundamental group of a non-orientable surface $M'_{g,h}$ is a free group with $g + h$ generators. The fundamental group of a non-orientable boundaryless surface M'_g has presentation

$$\langle a_1, \ldots, a_g \mid a_1^2 \ldots a_g^2 = 1 \rangle.$$

CHAPTER 3
ALGORITHMS FOR CURVES ON SURFACES

3.1 Introduction

Algorithmic problems for curves on surfaces have been studied already by Dehn in 1912 (see [Sti99]). Dehn gave algorithms for the CONTRACTIBILITY (deciding if a curve is null-homotopic, i. e., if it can be continuously contracted to a point) and TRANSFORMABILITY problems (deciding if a curve can be continuously transformed to another curve). Dehn's algorithms were based on the observation that the contractibility problem is equivalent to the word problem in $\pi_1(M)$ and the transformability problem is equivalent to the conjugacy problem in $\pi_1(M)$ (i. e., the (free) homotopy classes of a loop in M correspond to conjugacy classes of $\pi_1(M)$). Schipper [Sch92], Dey [Dey94] and Dey and Schipper [DS95] gave efficient algorithms for the contractibility problem based on the following observation [Poi95]: a curve α in M is null-homotopic iff its lift $\tilde{\alpha}$ in the universal covering space $\tilde{M}$ of M is closed. An optimal algorithm, based on results of Greendlinger [Gre60], for the transformability problem was given by Dey and Guha [DG99]. For two curves α_1, α_2 represented as edge-vertex sequences of length k_1, k_2 in a triangulation T, the algorithm decides if α_1 and α_2 are homotopic in time $O(|T| + k_1 + k_2)$.

The SIMPLICITY problem asks if a curve can be drawn without self-intersection (i. e, if there is a representative in the isotopy class which is a simple curve). A related DISJOINTNESS problem asks if two curves can be drawn in such a way that their drawings are disjoint. Reinhart [Rei62] gave an algorithm for the SIMPLICITY problem. Combinatorial algorithms for the problem were given by Zieschang [Zie65], Chillingworth [Chi69] and Birman and Series [BS84, BS87]. Chillingworth [Chi71] extended his algorithm to solve the DISJOINTNESS problem.

The Geometric Intersection Number problem is to determine the minimal number of intersections of two curves achievable by deforming the curves. Cohen and Lustig [CL87] gave the first combinatorial algorithm for computing the geometric intersection numbers on surfaces with non-empty boundary. The algorithm was extended by Lustig [Lus87] to the case of surfaces with no boundary. Another algorithm was given by Hamidi-Tehrani [HT97].

A Dehn twist along a simple closed curve γ in a surface is a homeomorphism of the surface to itself obtained by cutting the surface along γ, rotating one of the new boundaries 360 degrees (fixing everything outside a small neighborhood of the boundary) and then gluing the new boundaries back. The Dehn-Twist problem is to compute $D_\gamma(\alpha)$ for simple curves α, γ, where γ is closed. Penner [Pen84] gave explicit formulas for the action of the Lickorish generators of the mapping class group $\mathcal{M}$ on curves (or more generally measured laminations) represented by Dehn-Thurston coordinates. Hamidi-Tehrani and Chen [HTC96] gave an algorithm to compute the action of the Humphries generators of $\mathcal{M}$ on measured laminations represented by π_1-train tracks.

Our contribution

In Section 3.5 we give a polynomial-time algorithm to compute Dehn-twist of a simple curve along *any* simple closed curve where both curves are given in compressed representation. The previous algorithms ([Pen84, HTC96]) only allowed computing Dehn-twists along a *specific set* of simple closed curves.

In Section 3.7 we give the first polynomial-time algorithm for the Geometric Intersection Number problem for simple curves given by *compressed* representations. Note that the previous algorithms ([Rei62, CL87, Lus87, HT97]) worked over *explicit* representations and hence cannot be used for simple curves given by compressed representations.

3.2 Computing reduced words in a free group

In the Dehn-twist algorithm (section 3.5) we need to normalize a simple curve w.r.t. a triangulation. The normalization is equivalent to computing the reduction of a compressed word in a free group.

Let Σ be a set of generators of a free group and let $\Sigma^{\pm 1}$ be an alphabet. A word $w \in (\Sigma^{\pm 1})^*$ is **reduced** if a, a^{-1} are not adjacent in w for any $a \in \Sigma$. The **reduction** of a word $w \in (\Sigma^{\pm 1})^*$ is the unique word obtained by removing neighboring a, a^{-1} pairs until no such pair remains.

Lemma 3.2.1 *Let w be a word over $\Sigma^{\pm 1}$ given by a straight-line program P of length n. A straight-line program P' of the reduction w' of w can be computed in time $O(n^3)$ by a randomized algorithm running on a unit-cost RAM with word size $W = \Omega(n + \log \frac{1}{\delta})$. The probability of error is $\leq \delta$. The resulting straight-line program P' has length $O(n^2)$.*

Note that we can compute the cyclic reduction of a word w by computing reductions w', u' of w and $u = ww$ and then taking $w'[(1 + r)..(|w'| - r)]$ where $r = |w'| - |u'|/2$. Hence we have the following observation.

Corollary 3.2.2 *Let w be a word over $\Sigma^{\pm 1}$ given by a straight-line program P of length n. A straight-line program P' of the cyclic reduction w' of w can be computed in time $O(n^3)$ with probability $\geq 1 - \delta$ by a randomized algorithm running on unit-cost RAM with word size $W = \Omega(n + \log \frac{1}{\delta})$. The straight-line program P' has length $O(n^2)$.*

Let $\mathrm{SL}_2(K)$ be the group of unimodular 2×2 matrices over a commutative ring K. The **modular group** $\mathrm{PSL}_2(\mathbb{Z})$ is the factor group $\mathrm{SL}_2(\mathbb{Z})/(\pm I)$. There is a homeomorphism from $\mathrm{PSL}_2(\mathbb{Z})$ to $\mathrm{PSL}_2(\mathbb{Z}/p\mathbb{Z})$ obtained by reducing each entry modulo p (where p is a prime).

The modular group is a free product of finite groups generated by

$$S = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \quad \text{and} \quad R = \begin{pmatrix} 0 & 1 \\ -1 & 1 \end{pmatrix}.$$

Lemma 3.2.3 ([Imr84]) *Let B_m be the set of elements*

$$S^{a_1}(R^{a_2}S)\dots(R^{a_m}S)RSR(SR^{-a_m})\dots(SR^{-a_2})S^{a_1},$$

where $a_1 \in \{0,1\}$ and $a_2,\dots,a_m \in \{-1,1\}$. Elements of B_m are free generators of a free subgroup of $\mathrm{PSL}_2(\mathbb{Z})$. *Moreover, for any $X \in B_m$ we have $||X||_\infty \le 4^m$.*

Lemma 3.2.3 is useful for randomized equality testing of words (given by straight line programs) in the free group. Let $m = \lceil \log_2 |\Sigma| \rceil$. To each symbol $a \in \Sigma$ assign a distinct matrix $\psi(a)$ from the set B_m of Lemma 3.2.3. This defines a map $\psi : (\Sigma^{\pm 1})^* \to \mathrm{PSL}_2(\mathbb{Z})$. Let $\psi_p(x) = \psi(x) \bmod p$. Since B_m are free generators we have $\psi(w) = 1$ iff $w = \varepsilon$. Note that

$$||\psi(w)||_\infty \le 4^{m|w|}$$

and hence all non-zero entries in $\psi(w)$ have at most $2m|w|$ distinct prime divisors.

Observation 3.2.1 *Let $w \in (\Sigma^{\pm 1})^*$ be such that $w \ne 1$ in the free group. Let N be an integer and let p be a random prime between 1 and N. Then $\psi_p(w) \ne 1$ with probability at least*

$$1 - \frac{2m|w| \ln N}{N}.$$

Proof of Lemma 3.2.1:

Let N be an integer that will be specified later. We will work on a unit-cost RAM with word size $W = \Theta(\log N)$. Let p be a random prime between 1 and N. Note that we can multiply elements of $\mathrm{PSL}_2(\mathbb{Z}/p\mathbb{Z})$ in time $O(1)$.

W. l. o. g., we assume that P is binary straight-line program. Let $x_1,\dots,x_n$ be the variables in the straight-line program P. First we compute $\psi_p(x_i)$ for $i = 1,\dots,n$. This takes time $O(n)$ because we only need to perform one multiplication in $\mathrm{PSL}_2(\mathbb{Z}/p\mathbb{Z})$ for each assignment of P.

Claim: Let $y_1,\dots,y_k$ be the reductions of $x_1,\dots,x_k$. Suppose that we know the lengths of the y_i. Suppose that we are given $a, 1 \le a \le |y_k|$. In time $O(k)$ we can compute $\psi_p(y_k[1\dots a])$ and a' such that $y_k[1\dots a]$ is the reduction of $x_k[1\dots a']$.

Proof: Let $x_k := x_i x_j$ be the assignment to x_k in P. We have $y_k = y_i y_j$ in the free group. The equality is not necessarily true when the y_i are viewed as strings. Let $c = (|y_i| + |y_j| - |y_k|)/2$ be the length of the "cancellation" between y_i and y_j. There are two cases:

- If $a \leq |y_i| - c$ then we just need to compute $\psi_p(y_i[1 \dots a])$ (which is equal to $\psi_p(y_i[1 \dots a])$) and a' such that $y_i[1 \dots a]$ is the reduction of $x_i[1 \dots a']$.

- If $a > |y_i| - c$ we compute $\psi_p(x_i)\psi_p(y_j[1 \dots (a - |y_i| + c)])$ (which is equal to $\psi_p(y_i[1 \dots a])$). We also find a'' such that $y_j[1 \dots (a - |y_i| + c)]$ is the reduction of $x_j[1 \dots a'']$ and let $a' = |x_i| + a''$.

In both cases we spent only constant time and the index of the variable for which we need to recursively solve the problem decreased by at least 1. □

Suppose that we computed the lengths of the reductions $y_1, \dots, y_{k-1}$ of $x_1, \dots, x_{k-1}$. Let $x_k := x_i x_j$ be the next assignment in P. Using binary search and the claim we can find the length ℓ of the longest common prefix of y_j and y_i^{-1}. We take time $O(n^2)$ and perform $O(n)$ randomized equality tests. For now assume that all the equality tests did not err, we will analyze the total error probability later. Given ℓ we can compute $|y_k| = |y_i| + |y_j| - 2\ell$. Thus in time $O(n^3)$ we can compute $|y_i|$ for all $i = 1, \dots, n$.

We will generate the program for the reduced word $w' = y_n$ as follows. For k from n to 1 we do the following. Let $x_k := x_i x_j$ be an assignment in P. Let $c = (|y_i| + |y_j| - |y_k|)/2$ be the length of "cancellation" between y_i and y_j. Let a' be such that $y_i[1..(|y_i| - c)]$ is the reduction of $x_i[1 \dots a']$ and let b' be such that $y_j[(1+c)..|y_j|]$ is the reduction of $x_j[b' \dots |x_j|]$. In time $O(n)$ we can compute $a_1, \dots, a_m, b_1, \dots, b_{m'}$ such that $x_i[1 \dots a'] = x_{a_1} \dots x_{a_m}$ and $x_j[b' \dots |x_j|] = x_{b_1} \dots x_{b_{m'}}$ where $m, m' \leq n$. We replace $x_k = x_i x_j$ by $x_n = x_{a_1} \dots x_{a_m} x_{b_1} \dots x_{b_{m'}}$.

The resulting straight-line program has size $O(n^2)$ and it represents the reduction w' of w. The algorithm executes $O(n^2)$ randomized identity tests. Thus the total probability of failure is bounded by

$$\frac{2m|w|n^2 \ln N}{N}.$$

For $N = (1/\delta)4mn^2|w|\ln|w|$ the error probability is bounded by δ. We have $\ln N = O(n + \log\frac{1}{\delta})$. ■

Note that equality testing of words $u, v \in \Sigma^*$ given by straight-line programs can be reduced to the problem of computing a reduction of a word over $(\Sigma^{\pm 1})$ (since $u = v$ iff $uv^{-1} = \varepsilon$). The fastest known deterministic algorithm for equality testing takes time $O(n^4)$, if both the programs have size $O(n)$ [MST97]. Thus beating Lemma 3.2.1 by a deterministic algorithm would need an improvement for equality testing over monoids. We can compute reduction by a deterministic polynomial-time algorithm.

Lemma 3.2.4 *Let w be a word over $\Sigma^{\pm 1}$ given by a straight-line program P of length n. A straight-line program P' of the reduction w' of w can be computed in time $O(n^{10})$ by a deterministic algorithm running on unit-cost RAM with word size $W = \Omega(n)$. The straight-line program P' has length $O(n^2)$.*

Proof :

Now we sketch a deterministic algorithm for the problem. The algorithm considers the assignments in the order they occur in the program and computes straight-line program for the reductions $y_1, \ldots, y_n$ of $x_1, \ldots, x_n$.

Suppose that we already have a straight-line program P' for $y_1, \ldots, y_{k-1}$ of size t. Let $x_k := x_i x_j$ be the next assignment in P. We can compute the length of the reduction between y_i and y_j in time $O(n \cdot t^4)$ using binary search and the deterministic equality testing algorithm of [MST97]. In time $O(n^2)$ we can compute $a_1, \ldots, a_m, b_1, \ldots b_{m'}$ such that the reduction of $y_i y_j$ is $y_{a_1} \cdots y_{a_m} y_{b_1} \cdots y_{b_{m'}}$ and $m, m' \leq n$. Add assignment $y_k = y_{a_1} \cdots y_{a_m} y_{b_1} \cdots y_{b_{m'}}$ to P'. We increased the size of P' by $O(n)$. Hence during the course of algorithm $t = O(n^2)$. The total running time is $O(n^{10})$. ■

We have the deterministic analogue of Corollary 3.2.2. Unfortunately the running time is not practical.

Corollary 3.2.5 *Let w be a word over $\Sigma^{\pm 1}$ given by a straight-line program P of length n. A straight-line program P' of the cyclic reduction w' of w can be computed* *in time $O(n^{10})$ by an algorithm running on unit-cost RAM with word size $W = \Omega(n)$. The straight-line program P' has length $O(n^2)$.*

3.3 Curve coloring equations

Let M be a surface given by a pseudo-triangulation T. Let α be a simple multi-curve on M given by normal coordinates w.r.t. T. In this section we construct a quadratic system of word equations E with length constrains which will allow us to color the connected components of α. The elements of the alphabet Σ will be called colors.

First we arbitrarily pick an orientation for each edge e of the pseudo-triangulation T. Recall that e^{-1} stands for e with reversed orientation. We add a variable x_e to E which will list the colors of the segments of α intersecting e in the order they appear on e. Similarly $x_{e^{-1}}$ will list the colors of the segments of α intersecting e^{-1}. We add the following equation to E:

$$x_{e^{-1}} = x_e^R. \tag{3.1}$$

There are $\alpha(e)$ segments of α intersecting e and hence we add a length constraints

$$|x_{e^{-1}}| = |x_e| = \alpha(e). \tag{3.2}$$

Our goal now is to set the equations in E in such a way that segments which are adjacent on α get the same color (two segments s, t are adjacent is $s \cap s' \neq \emptyset$ and $s \cap s'$ does not intersect T). This will ensure that each component of α is monochromatic.

Let $t \in T$ be a triangle. Let e, f, g be the boundary edges of t such that efg is a cycle. There are three types of segments of α in t, depending on which two edges of t they connect. The numbers of each type $\ell_{ef}, \ell_{fg}, \ell_{ge}$ can be obtained by solving the following linear system of equations:

$$\begin{aligned} \ell_{ge} + \ell_{ef} &= \alpha(e) \\ \ell_{ef} + \ell_{fg} &= \alpha(f) \\ \ell_{fg} + \ell_{ge} &= \alpha(g). \end{aligned} \tag{3.3}$$

We add to E variables y_{ge}, y_{ef}, y_{fg} with constrains

$$|y_{ge}| = \ell_{ge}, \quad |y_{ef}| = \ell_{ef}, \quad |y_{fg}| = \ell_{fg}. \tag{3.4}$$

The variables y_{ge}, y_{ef}, y_{fg} list the colors for each type of segment listed in order from center of t towards its vertices. Finally we add following equations to E:

$$\begin{aligned} x_e &= y_{ge}^R y_{ef} \\ x_f &= y_{ef}^R y_{fg} \\ x_g &= y_{fg}^R y_{ge}. \end{aligned} \tag{3.5}$$

Definition 3.3.1 Let M be a surface given by a pseudo-triangulation T. Let α be a simple multi-curve given by normal coordinates w.r.t. T. The quadratic system of word equations (3.1) and (3.5) with length constrains (3.2) and (3.4) are called **curve coloring equations** of α.

The solutions of E correspond to colorings of the components of α. The number of free variables in the most general solution of E is the number of components of α. We can use the Diekert-Robson algorithm (Theorem 2.2.10) to compute the most general solution and then count the free variables.

Observation 3.3.1 *Let M be a surface given by a pseudo-triangulation T. Let α be a simple multi-curve given by normal coordinates w.r.t. T. Let n be the complexity of the normal coordinates of α. The number of connected components of α can be found in time $O(n)$ on a unit-cost word RAM with word size $W = \Omega(n)$.*

Next we would like to identify types (i. e., normal isotopy classes) of components of α and count how many copies of each type there are. The number of components of α can be exponential in the input size. However it is known that the number of types is small [Kne30]. We include a proof for completeness.

Lemma 3.3.2 *Let M be a surface given by a pseudo-triangulation T. Let α be a simple multi-curve on M normal w.r.t. T. The components of α fall into at most $6|T|$ normal isotopy classes.*

Proof :

Each triangle $t \in T$ is cut into regions by α. The central region and the regions which contain a vertex of t are called special. There are at most 4 special regions in each $t \in T$. There are at most 6 arcs in each triangle bordering a special region. Then there are at most $6|T|$ components of α which border a special region.

If a component C of α does not border a special region then it is normal isotopic to a component of α next to it. Therefore removing C from α does not decrease the number of normal isotopy classes of components of α. After we remove all components of α which do not border a special region we get a simple multi-curve with at most $6|T|$ components and the same number of normal isotopy classes of components as α.

Hence α has at most $6|T|$ normal isotopy classes. ■

Lemma 3.3.3 *Let M be a surface given by a pseudo-triangulation T. Let α be a simple multi-curve given by normal coordinates w.r.t. T. Let n be the complexity of the normal coordinates of α. The list of normal isotopy classes of components of α, together with their counts in α can be found in time $O(n^2|T|)$ on a unit-cost word RAM with word size $W = \Omega(n)$.*

Proof :

Pick an edge e in E_T which is intersected by α. We can color the i-th intersection of α with e by color b as follows. We replace x_e by ybz in E, where y, z are new variables with $|y| = i - 1$ and $|z| = \alpha(e) - i$. After solving E the numbers of b's in the x_e give the normal coordinates of the normal isotopy class containing the i-th intersection.

We compute the normal coordinates of the component C containing the 1-st intersection of α and e and then by binary search we find the largest k such that the component containing k-th intersection of α and e has the same normal coordinates as C. There are k copies of C in α. We take the curve $\alpha - kC$ and repeat. Lemma 3.3.2 implies that the number of repetitions is bounded by $6|T|$. ■

The curve coloring equations give a fast algorithm to decide whether two simple curves are normal isotopy disjoint.

Lemma 3.3.4 *Let M be a surface given by a pseudo-triangulation T. Let α, β be simple curves given by normal coordinates on T. Let m, n be the complexity of the coordinates of α, β. Deciding whether α, β are normal isotopy disjoint can be done in time $O(m + n)$ on a unit-cost RAM.*

Proof :
Let γ be the simple multi-curve with normal coordinates given by $\gamma(e) = \alpha(e) + \beta(e)$. Then α and β are normal isotopy disjoint iff γ has two components α and β. Set the curve coloring equations for γ and color one segment by color a. Let τ be the curve with normal coordinates $\tau(e) = \#_a(x_e)$. We only need to check whether $\tau = \alpha$ or $\tau = \beta$. ■

3.4 Compressed intersection sequence

Let M be a surface given by a pseudo-triangulation T. Let α be a simple curve on M. We can represent α by intersection sequence w_α with T. Unfortunately w_α is exponentially long in the complexity of the other representations (e. g., normal coordinates). In this section we will show that the intersection sequence can be compressed by a straight-line program. That motivates the following definition:

Definition 3.4.1 We say that a simple curve α on M is given by a **compressed intersection sequence** with T if we are given a straight-line program P for the intersection sequence w_α of α. The **complexity of the compressed intersection sequence** is the size of the program P.

If α given by compressed intersection sequence w_α is normal (i. e. w_α is reduced) then computing the normal coordinates of α is easy: we just need to count the number of occurrences of each $e \in E_T$ in w_α. From Lemma 2.2.3 we have:

Observation 3.4.1 *Let M be a surface given by a pseudo-triangulation T. Let α be a simple curve in M given by compressed intersection sequence with T of complexity n. Suppose that α is normal w.r.t. T. The normal coordinates of α can be computed in $O(n \cdot |T|)$ time on a unit-cost RAM with word-size $\Omega(n)$.*

Lemma 3.4.2 *Let M be a surface given by a pseudo-triangulation T. Let α be a simple curve in M given by normal coordinates of complexity n. The compressed intersection sequence w_α of α with T can be computed in $O(n \cdot |T|)$ time on a unit-cost RAM with word size $\Omega(n \cdot |T|)$. The complexity of the program for w_α is $O(n \cdot |T|)$.*

Proof :

First we assume that α is closed. We will create a quadratic system of word equations E with length constrains such that their solution will give a compressed intersection sequence w_α of α with T.

Let L be the sum of all coordinates of α. Note that any intersection sequence of α with T has length L.

We arbitrarily pick an orientation for each edge e of the triangulation T. For each $e \in E_T$ we add two copies e_l, e_r of e in M, one on each side of e. The copies are in a small neighborhood of e. We add variables x_{e_l} and x_{e_r} to E and add length constrains

$$|x_{e_l}| = |x_{e_r}| = L \cdot \alpha(e). \tag{3.6}$$

Now we describe what we want the solution of E to be. We view x_{e_l} and x_{e_r} as strings consisting of $\alpha(e)$ blocks of length L. The i-th block in x_{e_r} will contain the intersection sequence of α with T where the starting point is chosen to be the i-th intersection of α and e_r and the direction is to the left of e (i. e. the first edge of T to be intersected is e). Similarly the i-th block in x_{e_l} will contain the intersection sequence of α with T where the starting point is chosen to be the i-th intersection of α and e_l and the direction is to the left of e. We call x_{e_l} and x_{e_r} **concatenated intersection sequence** of α with T on e. Note that every block of x_{e_r} starts with e and every block of x_{e_l} ends with e.

We add following equation to E:

$$e x_{e_l} = x_{e_r} e \tag{3.7}$$

Let e' be e with the opposite orientation. We add variables $x_{e'_l}$ and $x_{e'_r}$ to E with the same length constraint as (3.6). Furthermore we add the following equations to

E:

$$x_{e'_R} = x_{e_L}^{-1} \text{ and } x_{e'_L} = x_{e_R}^{-1} \tag{3.8}$$

Let $t \in T$ be a triangle. Let e, f, g be the edges of T. W.l.o.g., the orientations of e, f, g are such that they form a cycle and that e_r, f_r, g_r lie inside t (the other case in which e_l, f_l, g_l lie inside t is analogous).

We add to E new variables y_{ge}, y_{ef}, y_{fg} with length constrains

$$|y_{ge}| = L \cdot \ell_{ge}, \quad |y_{ef}| = L \cdot \ell_{ef}, \quad |y_{fg}| = L \cdot \ell_{fg}. \tag{3.9}$$

where ℓ_{ge}, ℓ_{ef}, and ℓ_{fg} are defined by (3.3). Finally we add following equations to E:

$$\begin{aligned} x_{e_r} &= y_{ge}^{-1} y_{ef} \\ x_{f_r} &= y_{ef}^{-1} y_{fg} \\ x_{g_r} &= y_{fg}^{-1} y_{ge}. \end{aligned} \tag{3.10}$$

The equations (3.7), (3.10) as well as length constrains are satisfied by the intended solution. It remains to show that E has unique solution. Let $e \in E_T$. Let p be the first intersection point of α on e. We traverse α in the direction to the right of e. By induction we can show that in the i-th intersection of α with T, e occurs at i-th position in the appropriate segment of f_l or f_r. The base case is covered by (3.7).

By Theorem 2.2.10 the solution of E can be found in time

$$\sum_{x \in \Omega} \ln x = O\left(|T| \cdot \ln \left(\sum_{e \in E_T} \alpha(e) \right) + \sum_{e \in E_T} \ln \alpha(e) \right) = O(|T| \cdot n).$$

Now we deal with the case when α is a simple arc. If α connects edges $e, f \in E_T$ on ∂M then we add equation $x_e = x_f$ to E, effectively making the curve closed. ■

3.5 Computing Dehn-twists

In this section we show how normal coordinates of a Dehn twist of a simple curve α along a simple closed curve γ can be computed in time polynomial in the complexity of the normal coordinates of α and γ.

A **substitution system** S is a map $\Sigma^3 \to \Sigma^*$. Given a word $w = w_1 w_2 \dots w_n$ we let $S[w]$ be the word obtained by simultaneously applying substitutions $abc \mapsto aS(abc)bc$ to w, i. e.,

$$S[w] := w_1 S(w_1 w_2 w_3) w_2 S(w_2 w_3 w_4) w_3 \dots w_{n-2} S(w_{n-2} w_{n-1} w_n) w_{n-1} w_n.$$

Given a cyclic word $w = w_1 w_2 \dots w_n$ we let $S_c[w]$ be the word obtained by simultaneously applying substitutions $abc \mapsto aS(abc)bc$ to w, i. e.,

$$S_c[w] := w_1 S(w_1 w_2 w_3) w_2 S(w_2 w_3 w_4) w_3 \dots w_{n-1} S(w_{n-1} w_n w_1) w_n S(w_n w_1 w_2).$$

Lemma 3.5.1 *Let S be a substitution system given by a straight-line program R of size n. Let a cyclic word w be given by a straight-line program P of size m. A straight-line program P' for $S_c[w]$ can be computed in time $O(m)$ on a unit-cost RAM. The resulting program has size $O(m+n)$.*

Proof :

In $O(m)$ time we can compute for each x_i in P what the first two and the last two symbols of x_i are.

Let $S'[x_i]$ be $S[x_i]$ with the last symbol removed (i. e., $S'[x_i] = (S[x_i])[1...|S[x_i]|-1]$). We will build a straight-line program which computes $S'[x_i]$ for the x_i in P. Suppose that we already have a program P' which computes x_i and x_j. Let $x_k := x_i x_j$ be the next assignment of P. Let ab be the last two symbols of x_i and cd be the first two symbols of x_j. Then

$$S'[x_k] = S'[x_i] S(abc) b S(bcd) S'[x_j]. \tag{3.11}$$

The $S(abc)$ and $S(bcd)$ is already computed by R and hence (3.11) has only constant size.

Let x_n be the last variable. Let ab be the first two symbols of x_n and cd be the last two symbols of x_n. We have $S_c[x_n] = S'[X_n] S(cda) d S(dab)$. ■

Theorem 3.5.2 *Let M be a surface given by a pseudo-triangulation T. Let be α and γ be simple curves on M given by normal coordinates. Assume that γ is closed. Let n be an upper bound on the sizes of coordinates of α and γ. An intersection sequence of a curve β in the isotopy class of the Dehn twist $D^k_\gamma(\alpha)$ can be computed in time $O((n + \log k)|T|)$ by an algorithm running on unit-cost RAM with word size $W = \Omega((n + \log k)|T|)$.*

Proof :

Recall that each edge $e \in E_T$ is given orientation and e with orientation reversed is called e^{-1}. For each $e \in E_T$ we add two copies e_l, e_r of e in M, one on each side of e. The copies are in a small neighborhood of e.

The isotopy class of a Dehn twist $D_\gamma(\alpha)$ is independent of the choice of the representatives for α and γ. For each edge of $e \in T$ define three segments on e called left, middle and right window. We can deform α, γ so that

- γ always crosses e_l, e_r in the middle window (for all $e \in E_T$),
- α always crosses e_l, e_r in left or right window (for all $e \in E_T$), and
- all the crossings of α and γ happen in the disks bound by e_l and e_r, $e \in E_T$.

Now we create a substitution system S which will compute an intersection sequence of $D_\gamma(\alpha)$ with T. Let $e \in E_T$.

If e is bounded by the same triangle t then no intersections between α and γ occur between e_l and e_r. Now assume that e is bounded by two different triangles t_1 and t_2 (see Figure 3.1). Let e, f_1, g_1 be the boundary of t_1 and e, f_2, g_2 be the boundary of t_2. Let $u = f_1 e^{-1} g_2$, $v = f_2^{-1} e g_1^{-1}$. For each occurrence of u, v, u^{-1} and v^{-1} in w_α there are $\gamma(e)$ intersections between α and γ inside the disk bound by e_l and e_r (recall that $\gamma(e)$ is the number of intersections of γ with e). We add the following substitution to S:

$$\begin{array}{rcl} g_1 e^{-1} f_2 & \mapsto & g_1 (x_{e_l})^z e^{-1} f_2 \\ g_2^{-1} e f_1^{-1} & \mapsto & g_2^{-1} (x_{e_r})^z e f_1^{-1} \\ f_2^{-1} e g_1^{-1} & \mapsto & f_2^{-1} (x_{e_r})^{-z} e g_1^{-1} \\ f_1 e^{-1} g_2 & \mapsto & f_1 (x_{e_l})^{-z} e^{-1} g_2, \end{array} \tag{3.12}$$

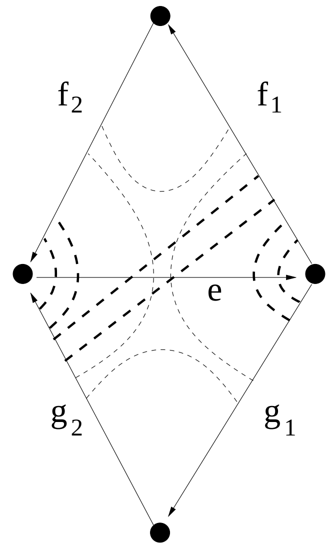

Figure 3.1: The drawing of α and γ used in computing the Dehn-twist of α along γ.

where $z = \pm 1$ is chosen depending on the orientation of γ. The sequence $w = S[w_\alpha]$ is an intersection sequence of $D_\gamma(\alpha)$ with T. ■

The curve β computed by the previous algorithm is not in normal position w.r.t. T (or in other words w is not reduced). To obtain a representative in normal position we need to reduce w using Lemma 3.2.1. Then using Observation 3.4.1 we compute normal coordinates of $D_\gamma(\alpha)$. Hence we have

Corollary 3.5.3 *Let M be a surface given by a pseudo-triangulation T. Let be α and γ be simple curves on M given by normal coordinates. Assume that γ is closed. Let n be an upper bound on the sizes of coordinates of α and γ. The normal coordinates of the Dehn twist $D^k_\gamma(\alpha)$ can be computed in time $O(((n+\log k)|T|)^3)$ by a randomized algorithm running on unit-cost RAM with word size $W = \Omega((n+\log k)|T| + \frac{1}{\delta})$. The probability of error is $\leq \delta$.*

Can the randomized algorithm be turned into a Las Vegas randomized algorithm?

Question 1 *Suppose that the algorithm of Corollary 3.5.3 on input $D_\gamma(\alpha)$ outputs β and on input $D^{-1}_\gamma(\beta)$ it outputs α'. Suppose that α' has the same normal coordinates* *as α. Is it possible that $D_\gamma(\alpha) \neq \beta$ (i. e. is it possible that both executions of the algorithm failed)?*

There is a generalization of the Dehn-twist operation called multiplication of curves (see [Luo01]).

Question 2 *Can two simple curves α, β given by normal coordinates be multiplied in polynomial time?*

3.6 Computing algebraic intersection numbers

Note that for the drawing of curves used in the proof of Theorem 3.5.2 it is easy to compute the sum of the signs of the intersections between e_l and e_r. For this we only need to compute the number of occurrences of u, v, u^{-1}, v^{-1} in w_α and the number of occurrences of e, e^{-1} in γ. By Lemma 2.2.4 this can be done in linear time and hence we obtain:

Observation 3.6.1 *Let M be a surface given by a pseudo-triangulation T. Let α, γ be simple multi-curves on M given by normal coordinates. Let m, n be the complexity of the coordinates of α, γ. The algebraic intersection number $\hat{i}(\alpha, \gamma)$ can be computed in time $O(m + n)$ on a unit-cost RAM with word size $O(m + n)$.*

3.7 Computing geometric intersection numbers

Once we can compute Dehn twists in polynomial time we can compute geometric intersection numbers in polynomial time. We will use following two results about geometric intersection numbers.

Theorem 3.7.1 ([FLP79]) *Let M be a surface. Let α, β, γ be simple curves on M. Suppose that γ is closed. Then*

$$n \cdot i(\alpha, \gamma) i(\gamma, \beta) - i(\alpha, \beta) \leq i(\alpha, D_\gamma^n(\beta)) \leq n \cdot i(\alpha, \gamma) i(\gamma, \beta) + i(\alpha, \beta). \tag{3.13}$$

Theorem 3.7.2 ([Luo01]) *Let M be a surface. Let α, β, γ be simple curves on M. Suppose that γ is closed. Then*

$$f(n) = i(\alpha, D_\gamma^n(\beta)) \tag{3.14}$$

is a convex function of $n \in \mathbb{Z}$.

If we know how to compute intersection numbers with a simple curve α then we can use it to compute intersection numbers of other curves.

Lemma 3.7.3 *Let M be a surface. Let α, β, γ be simple curves on M. Suppose that γ is closed. Let $n = 2i(\alpha, \beta)$. Then*

$$i(\gamma, \beta) = \frac{i(\alpha, D_\gamma^{n+1}(\beta)) - i(\alpha, D_\gamma^n(\beta))}{i(\alpha, \gamma)}.$$

Proof :

Let $A = i(\alpha, \gamma)i(\gamma, \beta)$ and $B = i(\alpha, \beta)$. We have $kA - B \leq f(k) \leq kA + B$ for any k. Let $n = 2B$.

Suppose that $f(n+1) - f(n) \leq A - 1$. Then by convexity

$$f(n+1) \leq (A-1)(n+1) + f(0) \leq (n+1)A - B - 1,$$

which contradicts $(n+1)A - B \leq f(n+1)$.

Suppose that $f(n+1) - f(n) \geq A + 1$. Then by convexity for $k \geq 0$

$$f(n+k) - f(n) \geq k(A+1) + f(n) \geq k(A+1) + nA - B = (n+k)A + k - B,$$

which contradicts $f(n+k) \leq (n+k)A + B$ for $k = 2B+1$. Hence $f(n+1) - f(n) = A$.
■

Lemma 3.7.4 *Let M be a surface given by a pseudo-triangulation T. Let β, γ be simple curves on M given by normal coordinates. Assume that α is closed. Then $i(\beta, \gamma)$ can be computed in polynomial time.*

Proof :

Let e be an edge of triangulation T which is intersected by γ. Note that $i(e,\gamma)$ is the normal coordinate $\gamma(e)$ of γ on e. Let $n = 2i(e,\alpha) = 2\alpha(e)$. Compute $\beta'' = D_\gamma^{n+1}(\beta)$ and $\beta' = D_\gamma^{n+1}(\beta)$. By Lemma 3.7.3 we have

$$i(\beta,\gamma) = \frac{\beta''(e) - \beta'(e)}{\gamma(e)}.$$

■

If both α and β are simple arcs then we cannot use Lemma 3.7.4 to compute $i(\alpha,\beta)$. We will first show how to compute $i(\alpha,\beta)$ if the endpoints of α are on different components of ∂M than the endpoints of β.

Given a simple arc α whose endpoints lie on different components D_1, D_2 of ∂M, let $C\alpha$ be the closed curve which is the boundary of the set of points within ε distance of $\alpha \cup D_1 \cup D_2$ for sufficiently small ε.

Observation 3.7.1 *$i(C\alpha,\beta) = 2i(\alpha,\beta)$.*

Proof :

Note that $i(\alpha, C\alpha) = 0$. Consider a drawing minimizing all the intersections. Cut M along $C\alpha$. Let M' be the component containing D_1 and D_2. Note that M' is a pair of pants and α connects D_1 and D_2. The drawing of β in M' contains of essential simple arcs with both boundary points on $C\alpha$. Each intersection of β and α corresponds to two intersections of β and $C\alpha$. ■

Given a simple arc α whose endpoints lie on the same component D of ∂M, let $C_1\alpha, C_2\alpha$ be the closed curves which are the boundaries of the set of points within ε distance of $\alpha \cup D$ for sufficiently small ε.

Observation 3.7.2 *$i(\alpha,\beta) = \max\{i(C_1\alpha,\beta), i(C_2\alpha,\beta)\}$.*

Proof :

Note that $i(\alpha, C_1\alpha) = i(\alpha, C_2\alpha) = 0$. Consider a drawing minimizing all the intersections. Let M' be the component containing D. Note that M' is a pair of pants and α

is essential simple arc connecting $C_1\alpha$ to $C_2\alpha$. The drawing of β consists of essential simple arcs with endpoints on $C_1\alpha \cup C_2\alpha$. Call the arc $i-j$ type if it connects $C_i\alpha$ to $C_j\alpha$. The $1-1$ type and $2-2$ type cannot occur simultaneously. Hence w. l. o. g., we can assume that the $1-1$ type does not occur. The $2-2$ type has 2 endpoints on $C_2\alpha$, 0 endpoints on $C_1\alpha$ and it intersects α twice. The $1-2$ type has 1 endpoint on $C_2\alpha$, 1 endpoint on $C_1\alpha$ and it intersects α once. Hence $i(C_1\alpha, \beta) \leq i(C_2\alpha, \beta) = i(\alpha, \beta)$. ■

If α and β share boundary then we first compute their geometric intersection number $i_f(\alpha, \beta)$ w.r.t. isotopy which can move the boundary. If the endpoints of α are on different components D_1, D_2 of ∂M and t is the number of endpoints of β that lie on $D_1 \cup D_2$ then

$$2i_f(\alpha, \beta) = \max\{i(C\alpha, \beta) - t, 0\}.$$

If the endpoints of α are on the component D of ∂M and t is the number of endpoints of β that lie on D then

$$i_f(\alpha, \beta) = \max\{i(C_1\alpha, \beta) - t, i(C_2\alpha, \beta) - t, 0\}.$$

To compute $i(\alpha, \beta)$ it remains to compute the number of intersections k of α, β in a small neighborhood of the shared boundary. The number k is determined by the number of twists of α and β around the boundary (which can be determined by observing the initial segment of w_α and w_β). Then we can compute $i(\alpha, \beta) = i_f(\alpha, \beta) + k$.

3.8 Removing internal vertices

Lemma 3.8.1 *Let M be a surface given by a pseudo-triangulation T. Assume $\partial M \neq 0$. In time polynomial in $|T|$ we can find a pseudo-triangulation T' of M such that T' has no internal vertices. Moreover for a curve α in M given by normal coordinates of complexity n w.r.t. T we can find normal coordinates of $\alpha' \sim \alpha$ w.r.t. T' in time polynomial in n on a unit-cost RAM.*

Proof :

Let $v \in V_T$ be an internal vertex. We will remove v from T and obtain a pseudo-triangulation T' with one less internal vertex. For a simple curve α we would like to compute $\alpha' \sim \alpha$. We will represent curves by compressed intersection sequences.

Note that v must be connected to another vertex $w \in V_T$ by an edge $e \in E_T$. Suppose that e is bounded by two triangles t_1, t_2. Let e, f_1, g_1 be the boundary of t_1 and e, f_2, g_2 be the boundary of t_2. We can contract t_1 and t_2 by gradually shortening e. Since v is an internal vertex, this will not cause any boundaries to touch. Let T' be the resulting triangulation. Let the edge arising from merging of f_i and g_i be called f_i in T'. Let w_α be the intersection sequence of a curve α in M w.r.t. T. We construct the intersection sequence of $\alpha' \sim \alpha$ in M w.r.t. T' as follows. Let eu be the intersection sequence of a cycle around v. We replace e and e^{-1} by u^{-1} and u in w_α (this corresponds to moving α across v), and then replace g_1 by f_1^{-1} and g_2 by f_2^{-1}. The resulting intersection sequence corresponds to $\alpha' \sim \alpha$.

Suppose that e is bounded by one triangle t. Let f be the remaining edge of T. The vertex v' between the copies of e on the boundary of t has degree 1. Thus t forms a disk on M with boundary f. Removing t and its edges from M we obtain T'. Let w_α be the intersection sequence of a curve α in M w.r.t. T. We construct intersection sequence of $\alpha' \sim \alpha$ in M w.r.t. T' as follows. We remove e, f, e^{-1}, f^{-1} from w_α. The resulting intersection sequence corresponds to $\alpha' \sim \alpha$.

We repeat the procedure until no internal vertices remain. Note that the operations of w_α can be achieved by adding constantly many assignments to the program for P. ∎

CHAPTER 4
THE STRING GRAPH PROBLEM

4.1 Introduction

In this chapter we will study the string graph problem. A graph G is a **string graph** if it is the intersection graph of a set of simple curves in the plane (i. e., if there exists a collection of simple curves $\{\gamma_v \mid v \in V_G\}$ such that $\gamma_v \cap \gamma_u \neq \emptyset$ iff $\{u, v\} \in E_G$). The String Graph Problem asks for an algorithm which decides whether a given graph is a string graph.

4.1.1 Origin of the problem

Benzer in [Ben59] investigated the topology of the fine genetic structure (how the subelements of genes are linked together). A region of the genetic structure is modeled by a topological space T. A simple mutation is modeled by a connected set of T. Using a recombination test it is possible to decide whether two mutations overlap or not. Hence we can construct the adjacency matrix M of the intersection graph of the mutations and try to deduce properties of T from M.

Benzer was mainly interested in the linear topology and was able to confirm the hypothesis that the topology of the fine genetic structure is linear. However he asked the following general question:

> *It would be an interesting mathematical problem to derive the characteristic feature of the matrix for each kind of topological space.*

Sinden [Sin66] raised the question of recognizing string graphs explicitly. He was trying to determine realizability of thin RC-circuits (Resistor-Capacitor circuits). In a thin RC-circuit the resistors are made by depositing long narrow strips of conductor and the capacitors are made by crossing two conductors and putting a dielectric between them at the place where they cross (thus there are three layers at the place of a crossing). Let C be a circuit consisting of conductors with capacitors between some pairs of the conductors. Two conductors cross in a thin RC-circuit for C iff there is a capacitor in C between them. If two conductors cross then they can cross more than once as the capacitance can be distributed over several crossings. Sinden [Sin66] asked which circuits are realizable as a thin RC-circuit. He considered a version in which the resistance aspect is ignored (pure capacitor circuits) which is precisely the string graph problem. Sinden's question was popularized by Graham [Gra76].

4.1.2 Finitely many intersections suffice

It was observed in [KGK86] that a string graph can be realized by a set of simple curves which intersect only finitely many times. We include the proof of this fact. For basic plane topology see section 2.3.2.

Lemma 4.1.1 *Let G be a string graph. Then there is a family of polygonal curves realizing G such that any two curves intersect only finitely many times. Moreover the curves are in general position (i.e. in no point more than two curves intersect and all the intersections are transversal).*

Proof :

Assume we have a string graph realized by a family of curves $(C_i)_{i\in V_G}$. Let

$$\varepsilon = (1/2)\min\{d(C_i, C_j); C_i \cap C_j = \emptyset\}.$$

Note that $\varepsilon > 0$, since all the curves are compact sets. For each (i, j) such that $C_i \cap C_j \neq \emptyset$ we select a witness point $p_{i,j} \in C_i \cap C_j$.

Choose δ according to Proposition 2.3.9 such that all polygonal δ-skeletons of the curve C_i are within Hausdorff distance $\varepsilon/2$ of C_i (for all $i \in V_G$). Fix such a polygonal δ-skeleton P_i for each curve C_i and include on it the $p_{i,j}$ for all j such that $C_i \cap C_j \neq \emptyset$.

Perturb each point in P_i, $i \in V_G$ and each witness point $p_{i,j}$ by a distance less than $\varepsilon/2$ so that all the points end up in a general position. This defines new polygonal arcs

P_i' for which $\text{dist}(P_i, P_i') < \varepsilon/2$, and hence $\text{dist}(P_i', C_i) < \varepsilon$ (for $i \in V_G$). If $C_i \cap C_j = \emptyset$, then $d(C_i, C_j) \geq 2\varepsilon$, hence $d(P_i', P_j') \geq d(C_i, C_j) - \text{dist}(C_i, P_i') - \text{dist}(C_j, P_j') > 0$, and therefore P_i' and P_j' do not intersect. If on the other hand $C_i \cap C_j \neq \emptyset$ then P_i' and P_j' share the witness point $p_{i,j}$.

There are finitely many segments on each P_i' and all the points are in general position. Hence there are finitely many intersections. It is easy to remove the self intersections of the P_i'. ■

The following alternative characterization of string graphs will be useful.

Lemma 4.1.2 *A graph G is a string graph iff there is a set of simply connected regions $R_i, i \in V_G$ such that $R_i \cap R_j \neq \emptyset$ iff $\{i,j\} \in E_G$.*

Proof :

By taking infinitesimal neighborhoods of the polygonal arcs P_i', $i \in V_G$ we obtain a set of regions R_i, $i \in V_G$ in the plane. Each region is homeomorphic to a disc and two regions R_i, R_j intersect iff P_i', P_j' intersect. Thus every string graph is an intersection graph of a set of simply connected regions of the plane.

Assume we have a set of regions R_v, $v \in V_G$ in the plane. For each intersecting pair R_i, R_j we pick a witness point $p_{i,j} \in R_i \cap R_j$. Let C_i be a polygonal curve connecting the witness points in R_i. Clearly C_i and C_j intersect iff R_i and R_j intersect. Thus string graphs are intersection graphs of discs in the plane. ■

4.1.3 A graph which is not a string graph

Sinden [Sin66] constructed a graph which is not string graph. Given a graph G, let G^* be the graph obtained from G by subdividing each edge by one point.

Lemma 4.1.3 ([Sin66]) *G^* is a string graph iff G is a planar graph.*

Proof :

Assume G is planar. Then by [Fár48] there is a straight-line embedding D of G in the plane. We construct a set of curves representing G^* as follows. Vertex $v \in V_G \subseteq V_{G^*}$ will be represented by a small circular arc centered at v which intersects exactly the edges going to v. A vertex $v \in V_{G^*} - V_G$ belongs to an edge $e \in E_G$. It will be represented by a segment of $e \in D$ which intersects arcs of both its endpoints. The resulting collection of simple curves realizes G^*.

Assume G^* is a string graph. Let $C_1, \ldots, C_{|V_{G^*}|}$ be a set of curves realizing G^*. For each $v \in V_G$ cut out a hole containing points within distance ε from C_v. For sufficiently small $\varepsilon > 0$ the holes are disjoint and there is a simple arc connecting holes u, v (for all $(u, v) \in E_G$). We can contract the holes to punctures and obtain a planar drawing of G. ■

From Lemma 4.1.3 it follows that $(K_5)^*$ is not a string graph. The smallest example of a graph which is not a string graph has 12 vertices [KGK86].

4.1.4 Graph classes which are string graphs

Several well-studied classes of graphs (e. g., interval graphs, permutation graphs, circular-arc graphs) are contained in the class of string graphs.

Sinden [Sin66] observed that a complete graph K_n is a string graph for any n. The following result of [KGK86] is a generalization of this observation.

Lemma 4.1.4 ([KGK86]) *Let G be a graph. Let $V_G = V_1 \cup \cdots \cup V_k$ be a decomposition of the vertex set into disjoint sets such that the graph induced by V_i is complete for each $i \in [k]$. Let G' be the graph obtained from G by contracting each V_i to a single vertex. If G' is planar then G is a string graph.*

Proof :

Let D be the drawing of G' in the plane. Let $i \in [k]$ be a vertex of G' and $v \in V_i$. Let $\gamma_1, \ldots, \gamma_t : [0, 1] \to \mathbb{R}^2$ be the curves in D corresponding to the edges of G adjacent to v. Let R_v be the ε-neighborhood of $\cup_{i=1}^{t} \gamma_i([0, 3/4])$. For small $\varepsilon > 0$ regions R_v and R_u intersect iff $\{u, v\} \in G$. Hence, by Lemma 4.1.2, G is a string graph. ■

Corollary 4.1.5 ([Sin66]) *A planar graph is a string graph.*

Corollary 4.1.6 ([KGK86]) *Let G be a graph. If $\chi(\overline{G}) \leq 4$ then G is a string graph.*

Using the 4-color Theorem [AH77] we obtain the following nice observation.

Corollary 4.1.7 ([KGK86]) *A complement of a planar graph is a string graph.*

4.1.5 AT-graph realizability

An **abstract topological graph** (**AT-graph**) is a graph G with a set $R_G \subseteq \binom{E_G}{2}$. An AT-graph (G, R_G) is **realizable** if there is a drawing of G in the plane such that the curves C_e, C_f intersect iff $(e, f) \in R_G$, (for all $e, f \in E_G$). Note that pairs in R_G *must* intersect. The problem of deciding whether an abstract topological graph is realizable is called the AT-Realizability problem.

The following two results show that String Graph Recognition and AT-Realizability are polynomially equivalent.

Lemma 4.1.8 ([Kra91]) *Given a graph G we can construct in polynomial time an AT-graph H such that G is a string graph iff H is realizable. We have $|V_H| = 2|V_G|$, $|E_H| = |V_G|$ and $|R_H| = |E_G|$. If H is realizable with c intersections then G is realizable with c intersections.*

Proof [Kra91]:

Let G be a graph. Let H be a graph consisting of $|V_G|$ disjoint edges. Each edge corresponds to a vertex of G. We ask that edges $e, f \in E_H = V_G$ intersect iff $\{e, f\} \in E_G$. Clearly H is realizable iff G is a string graph. ■

Lemma 4.1.9 ([Kra91]) *Given an AT-graph H we can construct in polynomial time a graph G such that G is a string graph iff H is realizable. We have $|V_G| = |V_H| + 3|E_H|$ and $|E_G| = 4|E_H| + |R_H|$. If G is realizable with c intersections then H is realizable with $\leq 4c$ intersections.*

Proof [Kra91]:

Let H be an AT-graph. Let G be the following graph.

$$\begin{aligned} V_G &= V_H \cup E_H \cup \{(v, e); v \in e \in E_H\}, \\ E_G &= \{\{v, (v, e)\}, v \in e \in E_H\} \cup \{\{e, (v, e)\}, v \in e \in E_H\} \cup R_H. \end{aligned}$$

Suppose that H is realizable. Let $C_{(v,e)}$ be the initial part of $e = (v, u)$. Let C_e be the middle part of e intersecting both initial parts on e. Finally let C_v be a small circular arc around v intersecting only the initial parts of edges adjacent to v. G is the intersection graph of the curves C and hence G is a string graph.

Assume that G is a string graph. Let $C = \{C_v; v \in G\}$ be a set of curves realizing G. We can assume (by Lemma 4.1.1) that C is in general position. For each $v \in e \in E_H$ we will do the following operation. On $C_{(v,e)}$ pick neighboring intersection points a_v and a_e with C_v and C_e. Connect one endpoint of C_e to a point a'_v between a_v and a_e by a curve $\beta_{e,v}$ running close to C_e. Merge curves $\beta_{e,v}$ and C_e. Remove parts of $C_{(v,e)}$ beyond a'_v and a_v. Note that the curves still realize H. Now we have $|C_{v,e} \cap C_v| = 1$ and $|C_{v,e} \cap C_e| = 1$.

For each $v \in V_H$ contract C_v and $C_{(v,e)}$ to a single point. The resulting drawing realizes H. The number of intersections has at most quadrupled (because the $\beta_{e,v}$ double parts of the e). ■

An AT-graph (G, R_G) is **weakly realizable** if there is a drawing of G such that if curves C_e, C_f intersect then $(e, f) \in R_G$ (i. e., the pairs in R_G *may* intersect). The problem of deciding weak realizability of an AT-graph is called AT-Weak Realizability. The following lemma shows that AT-Weak Realizability is at least as hard as String Graph Recognition and AT-Realizability.

Lemma 4.1.10 ([MNT88]) *Given a graph G we can construct in polynomial time an AT-graph H such that G is a string graph iff H is weakly realizable. We have $|V_H| = |V_G| + |E_G|$ and $|E_H| = 2|E_G|$. If H is weakly realizable with c intersections then G is weakly realizable with $\leq 4c + |E_G|$ intersections.*

Proof :
Let $V_H = V_G \cup E_G$ and $E_H = \{\{v, e\}; v \in e \in E_G\}$. Two edges $\{u, e\}$ and $\{v, f\}$ are allowed to intersect if $\{u, v\} \in E_G$.

Assume that H is weakly realizable. Let R_v be be an ε-region of the edges going from v. Then two regions R_v, R_u may intersect only if u, v are connected by an edge in G. If u, v are connected by an edge e in G then the regions R_u, R_v intersect. Hence the regions $R_v,\ v \in V_G$ realize the string graph G.

Assume that G is a string graph. Let D be a realization of G. Let $C_v,\ v \in V_G$ be a set of curves realizing G. For each $\{u, v\} \in G$ pick a point $p_{u,v} \in C_v \cap C_u$. Let p_u be an endpoint of C_u. Connect p_u to all the $p_{u,v}$ with disjoint arcs running close to C_u. The obtained topological graph weakly realizes H. ∎

Kratochvíl [Kra91] reduced the Planar 3-Connected 3-Satisfiability to AT-Realizability.

Theorem 4.1.11 ([Kra91]) AT-Realizability *is* NP*-hard.*

Lemmas 4.1.9 and 4.1.10 yield NP-hardness for the other two problems.

Corollary 4.1.12 ([Kra91]) AT-Weak Realizability *and* String Graph Problem *are* NP*-hard.*

4.1.6 Exponential obstacle

Kratochvíl and Matoušek [KM91] have shown that there is an AT-graph G such that any realization of G has exponentially many intersections.

Theorem 4.1.13 ([KM91]) *There is a realizable AT-graph G with $2n+3$ vertices and $4n+11$ edges such that in any realization of G there are at least $2^n - 1$ intersections on one of the edges.*

Lemma 4.1.9 implies that the same obstacle exists for string graphs.

Corollary 4.1.14 *There is a string graph G with $14n + 36$ vertices such that any realization has at least 2^{n-2} crossings.*

Proof of Theorem 4.1.13 [KM91]:
Let H be the graph on Figure 4.1. The graph H is planar and has a topologically unique embedding in the plane. Let G be the graph obtained from H by adding edges $\{u_i, v_i\},\ i = 1, \ldots, n$. The edge $\{u_i, v_i\}$ must cross edge e and the edges

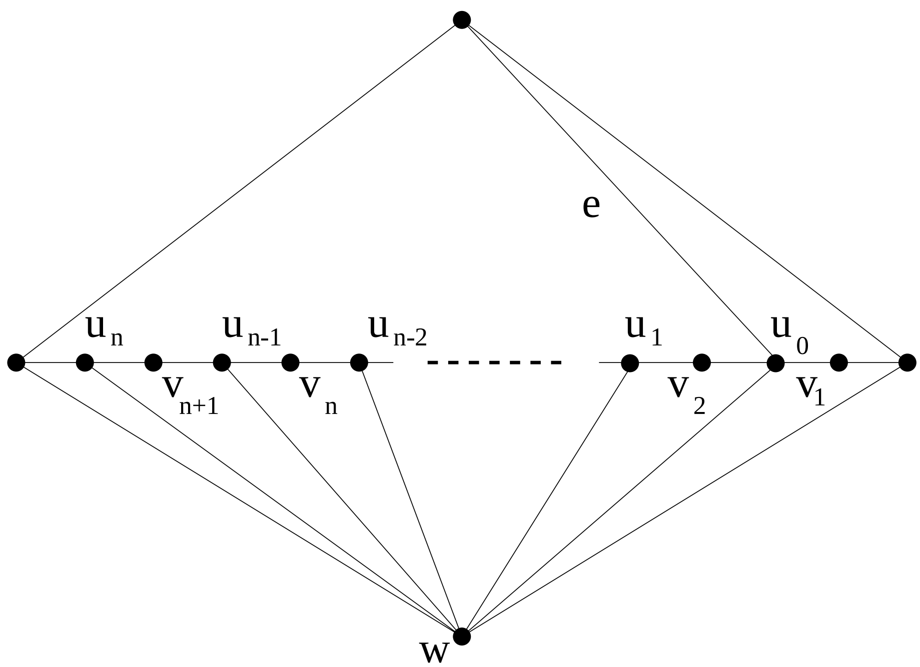

Figure 4.1: The graph H used in the construction of an AT-graph requiring exponentially many intersections.

$\{v_k, u_{k-1}\}, \{v_k, u_{k-2}\}$ for k such that $1 \leq k < i$. No other pairs of edges may intersect.

We prove by induction on $i = 1, \ldots, n$ that in any drawing of G_T, $\{u_i, v_i\}$ has to cross e at least 2^{i-1} times. The claim is clearly true for $i = 1$. Assume that it is true for $i < k$. Delete the edges $\{u_i, v_i\}, i > k$ from the drawing. Let C be the circle $w, u_{k-1}, v_{k-1}, u_{k-2}, w$. None of the edges of C can intersect and hence C is embedded. The circle C separates u_k and v_k and hence $\{u_k, v_k\}$ must intersect $\{v_{k-1}, u_{k-2}\}$ because it is the only edge on C that $\{u_k, v_k\}$ can intersect. Let a be an intersection point of $\{u_k, v_k\}$ and $\{v_{k-1}, u_{k-2}\}$.

Any arc connecting u_k to a can be made to connect $\{u_{k-1}, v_{k-1}\}$ without introducing any new intersections. Similarly any arc connecting v_k to a can be made to connect $\{u_{k-1}, v_{k-1}\}$ without introducing any new intersections. Hence by induction hypothesis the total number of intersections of $\{u_k, a\}, \{v_k, a\}$ with e is at least 2^k. ■

4.2 Connectivity preserving redrawing

4.2.1 The shortcut problem

Let M be a surface of genus g. Let $\alpha, \gamma_1, \ldots, \gamma_k$ be a collection of simple arcs on M. Suppose that α is intersected n times by the γ_i, i. e., $\sum_{i=1}^{k} i(\alpha, \gamma_i) = n$. We say that a collection of simple arcs $\alpha, \beta_1, \ldots, \beta_k$ **simplifies** the collection $\alpha, \gamma_1, \ldots, \gamma_k$ if

- the β_i connect the same endpoints as the γ_i, (i. e., $\partial\beta_i = \partial\gamma_i,\ i \in [k]$);
- the pairwise intersection numbers of the β_i do not increase, (i. e., $i(\beta_i, \beta_j) \leq i(\gamma_i, \gamma_j),\ i, j \in [k]$); and
- the total number of intersections with α decreases, (i. e., $\sum_{i=1}^{k} i(\alpha, \beta_i) < n$).

For example note that if $k = 1$ and $n \geq 2$ then the collection α, γ_1 can always be simplified.

Question 3 *Let k, g be non-negative integers. Let M be a surface of genus g. Does there exist $n_0 = n_0(k, g)$ such that any collection $\alpha, \gamma_1, \ldots, \gamma_k$ on M with*

$$\sum_{i=1}^{k} i(\alpha, \gamma_i) \geq n_0$$

can be simplified?

One approach to question 3 is to try to redraw the γ_i only in a small neighborhood of α. Take a neighborhood D of α in which the collection $\alpha, \gamma_1, \ldots, \gamma_k$ is homeomorphic to the following picture: the neighborhood is a unit disc, α is the x axis (restricted to the disc) and each $\gamma_i \cap D$ is a collection of vertical segments in D. We say that a collection $\alpha, \beta_1, \ldots, \beta_k$ simplifies $\alpha, \gamma_1, \ldots, \gamma_k$ **locally** if no new routes are introduced outside of D (i. e., $\beta_i \cap \overline{D} \subseteq \gamma_i \cap \overline{D}$). Let $\Upsilon(g, k)$ be the smallest integer such that any collection of simple arcs $\alpha, \gamma_1, \ldots, \gamma_k$ with $\sum_i i(\alpha, \gamma_i) \geq \Upsilon(g, k)$ can be locally simplified.

Question 4 (Shortcut problem) *What is the value of $\Upsilon(g, k)$?*

In the simplest version of the shortcut problem the γ_i are pairwise disjoint. Let $\Upsilon_s(g,k)$ be the smallest integer such that any collection of simple arcs $\alpha, \gamma_1, \ldots, \gamma_k$ with disjoint γ_i (i. e., $i(\gamma_i, \gamma_j) = 0$ for $i, j \in [k]$) and $\sum_i i(\alpha, \gamma_i) \geq \Upsilon_s(g,k)$ can be locally simplified. Note that $\Upsilon(g,k) \geq \Upsilon_s(g,k)$.

Question 5 (Simple shortcut problem) *What is the value of* $\Upsilon_s(g,k)$*?*

We will determine the value of $\Upsilon(g,k)$ and $\Upsilon_s(g,k)$ in the case $g = 0$. We conjecture that there is a bound on $\Upsilon_s(g,k)$ independent of g.

Conjecture 2 *There exists a constant* C *such that for all* k, g*:*

$$\Upsilon_s(g,k) \leq C^g.$$

The bounds on $\Upsilon_s(g,k)$ shown in table 4.1 were computed by a program. Figures 4.2 and 4.3 show example configurations which cannot be reduced. The numbers in the middle row indicate to which curve a vertical segment belongs. For each curve α we number the intersection points with ∂D in the order they occur on α. These are the numbers in top and bottom rows. The reader can easily (i. e., in polynomial time) check that such a configuration can occur on a surface of genus 1. Checking whether a configuration is reducible seems harder and we don't have a polynomial time algorithm.

$g \diagdown k$	1	2	3	4	5	$\ldots$	k
0	2	4	8	16	32	$\ldots$	2^k
1	2	7	16	≥ 39			
2	2	7	≥ 27				
3	2	7					
$\vdots$	$\vdots$	$\vdots$					
∞	2	7	≥ 32				

Table 4.1: Known bounds for $\Upsilon_s(g,k)$.

```
    2   4   4   1   1   3
    |   |   |   |   |   |
 -  0 - 1 - 0 - 1 - 0 - 1-
    |   |   |   |   |   |
    3   5   5   0   0   2
```

Figure 4.2: An intersection pattern which cannot be locally simplified, showing $\Upsilon_s(1,2) \geq 7$.

```
   2  10   6   2   0   1   0   8   5   7   9  13   5   3   5
   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
 - 0 - 1 - 0 - 1 - 2 - 1 - 0 - 1 - 2 - 1 - 0 - 1 - 0 - 2 - 1-
   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
   3  11   7   3   1   0   1   9   4   6   8  12   4   2   4
```

Figure 4.3: An intersection pattern which cannot be locally simplified, showing $\Upsilon_s(1,3) \geq 16$.

4.2.2 The genus zero case of the shortcut problem

We can determine the values of $\Upsilon(g,k)$ and $\Upsilon_s(g,k)$ when the curves are drawn on a sphere with holes. An example of [KM91] shows $\Upsilon_s(0,k) \geq 2^k$ (see Lemma 4.1.13). The example is a collection of simple arcs $\alpha, \gamma_1, \ldots, \gamma_k$ such that the γ_i are pairwise disjoint, $\sum i(\alpha,\gamma_i) = 2^k - 1$, and the collection cannot be locally simplified. The example together with Lemma 4.2.2 yields.

Theorem 4.2.1

$$\Upsilon(0,k) = \Upsilon_s(0,k) = 2^k.$$

Lemma 4.2.2 ([SŠ04])

$$\Upsilon(0,k) \leq 2^k.$$

Proof :

Let $\alpha, \gamma_1, \ldots, \gamma_k$ be a collection of simple arcs on a sphere with h holes. Suppose that α is intersected at least 2^k times.

Claim: There exists a non-trivial segment S on α such that every γ_i intersects S even number of times.

Proof: If we cut α along the γ_i we get at least 2^k+1 connected components. Put a special point in each connected component. To each special point assign a vector in $\mathbb{Z}_2^k$ whose i-th coordinate is the parity of the number of intersections of γ_i with the part of α to the left of that special point. By pigeon-hole principle there are two special points labeled by the same vectors in $\mathbb{Z}_2^k$. The segment between these two special points is intersected even number of times by each γ_i. □

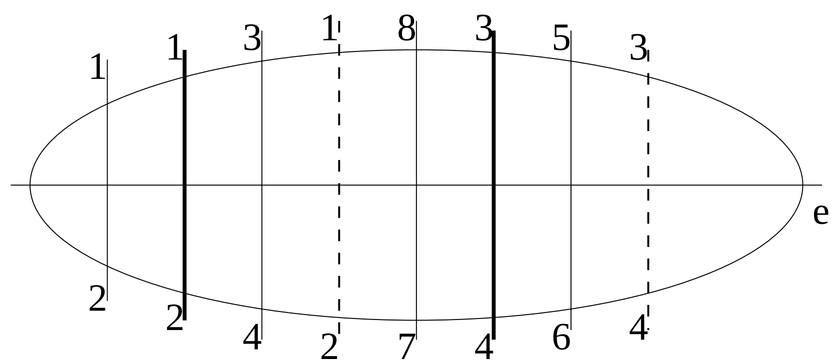

Figure 4.4: Segment S intersected an even number of times by each edge.

Let D be a neighborhood of S such that the drawing of the curves in D is homeomorphic to the following picture: the neighborhood is a unit disc, S is the x axis (restricted to the disc) and each $\gamma_i \cap D$ is a collection of vertical segments in D.

An intersection of a γ_i with ∂D will be called a port. For each arc γ_i pick an orientation and assign numbers $1, 2, \ldots, 4n_i$ to the ports of γ_i in the order they appear on γ_i (here $2n_i$ is the number of intersections of γ_i with S). The parts of γ_i connecting port $4j-2$ to port $4j-1$, $j \in [n_i]$ are called odd. The parts of γ_i connecting port $4j$ to $4j+1$, $j \in [n_i - 1]$ and the parts connecting ∂M to port 1 and $4n_i$ are called even. Note that both even and odd parts lie outside of D.

A mirror flip along the x-axis switches ports $2j-1$ and $2j$. Let h be the circular inversion followed by mirroring along the x-axis. Note that h applied to an odd part connecting port $4j-2$ to port $4j-1$ yields a connection of ports $4j-3$ and $4j$ inside D.

Let β_i be the union of the even parts of the γ_i and h(odd parts of the γ_i). Note that β_i is an arc that connects the same endpoints as γ_i because the even parts connect port $4j$ to $4j+1$, $j \in [n_i - 1]$ and the image of odd parts connects $4j-3$ to $4j$, $j \in [n_i]$. Moreover $i(\beta_i, \beta_j) \le i(\gamma_i, \gamma_j)$ because the intersections in D between images of odd parts were present in the drawing of the γ_i.

The number of intersections of β_i with ∂D is only $2n_i$ and hence $i(\alpha, \beta_i) \le i(\alpha, \gamma_i) - n_i$ because we can redraw e to be either the upper arc of ∂D, or the lower arc of ∂D. ■

4.3 String graph problem in NP

Now we are ready to show that AT-Weak Realizability has small witnesses.

Theorem 4.3.1 ([SSŠ02]) AT-Weak Realizability *is in* NP.

This together with results of Kratochvíl [Kra91] completely resolves the complexity of the string graph problem.

Corollary 4.3.2 AT-Weak Realizability, AT-Weak Realizability, *and* String Graph Problem *are* NP-*complete.*

Proof of Theorem 4.3.1:

Let (G, R) be an instance of the AT-Weak Realizability problem. Let n be the number of vertices and m be the number of edges of G. Suppose that (G, R) is weakly realizable. Let D be a weak realization of G. We drill holes around the vertices of G and triangulate the resulting surface M by a triangulation T. Note that M is a sphere with n holes. We obtained a collection of simple arcs on M such that if arcs α_e, α_f intersect then $\{e, f\} \in R$. Hence

$$i(\alpha_e, \alpha_f) > 0 \implies \{e, f\} \in R. \tag{4.1}$$

Conversely if we have a collection of simple arcs on M such that (4.1) is satisfied we can obtain a weak realization of G by taking representatives of the arcs which intersect minimally (Lemma 2.3.8) and then collapse the holes of M. It remains to show that there is a collection witnessing weak realizability which can be efficiently encoded.

Let $\alpha_e,\ e \in E_G$ be a collection of simple arcs satisfying (4.1) and such that

$$\mu = \sum_{g\in E_T} \sum_{e\in E_G} \alpha_e(g)$$

is minimal. If there is an edge $g \in E_G$ with $\sum_{e\in E_G} \alpha_e(g) \geq 2^m$ then, by Lemma 4.2.2, the α_e can be locally simplified, a contradiction with minimality of μ. Hence the complexity of normal coordinates of the α_e is $O(|T| \ln 2^m) = O(mn)$. We can verify in time polynomial in mn that for $\{e, f\} \notin R$ we have $i(\alpha_e, \alpha_f) = 0$ (using Lemma 3.3.4). ■

4.3.1 String graphs on other surfaces

A natural extension of the string graph problem is to consider intersection graphs of curves on surfaces of higher genus. In proving Theorem 4.3.1 there was only one place were we used the fact that the surface was a plane, namely when we applied the Lemma 4.2.2. We do not have the analogue of Lemma 4.2.2 for other surfaces. If Conjecture 2 is true then the string graph problem is in NP for higher genus surfaces.

In the absence of an analogue of Lemma 4.2.2 for any surface other than the plane this means that we face the question of decidability anew. This time we use a different approach suggested by the proof of Theorem 4.3.1. We will use the connection between weak realizability and equations on words.

Theorem 4.3.3 *The generalized* String Graph Problem, AT-Realizability *and* AT-Weak Realizability *are in* PSPACE *on any surface M.*

Proof :

We will show that AT-Weak Realizability is in PSPACE. Let (G, R) be an AT-graph. Let $\Sigma = E_G$. We will work over the monoid given by the graph (E_G, R_G) (i. e., the edges that can cross commute).

Let n be the number of vertices and m be the number of edges of G. Let M' be M with n holes. Assume that M' is given by pseudo-triangulation T.

As in the proof of 4.3.1 it is enough to find simple arcs α_f, $f \in E_G$ such that (4.1) is satisfied and that for $f = (u, v)$, α_f connects holes u, v. For each hole v we can guess the order in which the simple arcs appear on v. This gives a word $w_v \in \Sigma^*$. Now we consider the curve coloring equations given by (3.1) and (3.5) without length constrains. For edge e_v on the boundary of v we set $x_{e_v} = w_v$.

There is a correspondence between drawings in which forbidden pairs do not cross and solutions of the curve coloring equations over monoid given by (E_G, R_G). Hence we can use 2.2.11 to decide if the AT-graph (G, R) is weakly realizable. ■

4.4 Topological inference problem

As we noted before (Lemma 4.1.2) the string graph problem is equivalent to the following problem: for a collection of discs we specify which pairs must intersect and which may not, can the discs be drawn in the plane to fulfill these requirements?

In topological inference we look at finer relationships between discs. There are eight possible relationships of two discs based on whether the intersection of their interior, boundary and exterior is empty or not [Ege91]. For any two discs A, B exactly one of the relations is true.

$$
\begin{aligned}
\text{disjoint}(A,B) &\equiv A \cap B = 0 \\
\text{equal}(A,B) &\equiv A = B \\
\text{inside}(A,B) &\equiv A \subseteq B^\circ \\
\text{contains}(A,B) &\equiv \text{inside}(B,A) \\
\text{covered}(A,B) &\equiv A^\circ \subsetneq B^\circ \ \wedge\ \partial A \cap \partial B \neq \emptyset \\
\text{cover}(A,B) &\equiv \text{covered}(B,A) \\
\text{meet}(A,B) &\equiv A^\circ \subseteq \mathbb{R}^2 - B \ \wedge\ \partial A \cap \partial B \neq \emptyset \\
\text{overlap}(A,B) &\equiv A^\circ \cap B^\circ \neq \emptyset \ \wedge\ (\mathbb{R}^2 - A) \cap (\mathbb{R}^2 - B) \neq \emptyset
\end{aligned}
$$

We call a Boolean combination of the topological predicates a **topological expression**. A topological expression is **explicit**, if it specifies the relationship between any pair of variables, meaning it is of the form $\bigwedge_{A,B \in I} \mathcal{P}_{A,B}(A, B)$, where I is the set of variables, and $\mathcal{P}_{A,B}$ is one of the eight basic predicates (for each $A, B \in I$). We can always assume that the expression does not contain the predicates contains or cover, because we can substitute them by inside and covered. Quantifying topological expressions we obtain **topological formulas**. Determining the truth of these (where the universe is the set of all regions in the plane) is the goal of topological inference [GPP95]. Of main interest are the purely existential formulas, since they express the existence of diagrammatic representations of logical relationships (Euler diagrams). In this case we also speak of the **realizability** of a topological expression.

In some special cases realizability of topological expressions is in P. Planar map graphs were introduced in [CGP98]. **k-planar map graphs** are intersection graphs of discs with disjoint interiors such that at most k regions meet at a point. Planar graphs are exactly 3-planar map graphs. A graph is planar map graph if it is k-planar map graph for some k. The problem of recognizing planar map graphs is equivalent to realizability of explicit topological expressions containing only meet and disjoint relations. In [CGP98] a polynomial time algorithm for recognizing 4-planar map graphs was shown. In [Tho98] a polynomial time algorithm for recognizing planar map graphs was announced.

4.4.1 Topological reasoning is in NP

In this section we will show that the realizability of topological expressions is in NP. We will use weak realizability problem as a black box.

Talking about a realization of meet(A, B), or covered(A, B) we call any point belonging to $\partial A \cap \partial B$ a **contact point** of A and B. In the other cases points belonging to the intersection of ∂A and ∂B we call **intersection points**.

We will now show how to redraw a realization of an explicit topological expression to bound the number of contact points in the drawing. Note that for any explicit expression there is always an equivalent explicit expression not containing equal.

Lemma 4.4.1 *Let φ be an explicit topological expression not containing* equal*. If there is a drawing realizing φ, then there is a drawing realizing φ in which the number of contact points on each boundary is bounded by the square of the number of variables in φ.*

Proof :

Let $A_1, \ldots, A_{|I|}$ be the family of variables occurring in φ. We can assume that the variables are sorted such that for $i < j$ there is no covered(A_j, A_i). If such an ordering does not exist, then φ has no realization. For each meet(A, B) and covered(A, B) in φ we choose a witness point $p_{A,B} \in \partial A \cap \partial B$. Let

$$\begin{aligned} \varepsilon_1 &= \min\{d(\partial A, \partial B) : \varphi \text{ contains } \text{disjoint}(A, B), \text{ or } \text{inside}(A, B)\} \\ \varepsilon_2 &= \min\{\text{dist}(A \cap \partial B, \partial A) : \varphi \text{ contains } \text{overlap}(A, B)\} \end{aligned}$$

Note that $\varepsilon_1 > 0$, since the considered boundaries are closed and disjoint. Also $\varepsilon_2 > 0$, since there is a point in $A \cap \partial B$ which is inside A. Let $\varepsilon = 1/2 \min\{\varepsilon_1, \varepsilon_2\}$. If B is a region with $\text{dist}(B, A_i) \leq \varepsilon$ then

$$\begin{array}{cc} \text{inside}(A_i, A_j) \Rightarrow \text{inside}(B, A_j), & \text{inside}(A_j, A_i) \Rightarrow \text{inside}(A_j, B), \\ \text{disjoint}(A_i, A_j) \Rightarrow \text{disjoint}(B, A_j), & \text{overlap}(A_i, A_j) \Rightarrow \text{overlap}(B, A_j). \end{array} \tag{4.2}$$

Unfortunately the same is not true for meet and covered. We will redraw the regions one by one, removing unnecessary contact points while preserving the meet and covered relationships.

Suppose then that for $A_1, \ldots, A_{i-1}$ the only contact points on their boundaries are witness points. We will show how to redraw A_i to make this true for $A_1, \ldots, A_i$ while preserving that $A_1, \ldots A_{|I|}$ realize φ.

Let $\psi : D \mapsto A_i$ be the homeomorphism of the unit closed disc to A_i. Using the Jordan-Schoenflies Theorem we extend ψ to a homeomorphism of the whole plane to itself which we call ψ again. Since ψ is uniformly continuous, there exists η such that if $(1 - \eta)D \subseteq E \subseteq D$ then $\text{dist}(\psi(E), A_i) < \varepsilon$. Let F be the union of $\psi^{-1}(A_j)$ for which there is covered(A_j, A_i). By the induction hypothesis assumption $F \cap \partial D$

contains only witness points. Choose E such that $F \cup (1-\eta)D \subseteq E \subseteq D$ and E intersects ∂D only in witness points. Replace A_i by $\psi(E)$. By the implications in (4.2) all inside, disjoints and overlaps are preserved. Because E contains all witness points for region A_i all covered and meet relations are satisfied, and only the witness points are contact points of A_i. Since contact points of A_j, $j < i$ did not change, this will be true after redrawing all regions. ■

Before we prove the main result we need to introduce a variant of realizability. Let (G, R, S) be such that $R, S \subseteq \binom{E}{2}$, and $R \cap S = \emptyset$. We call (G, R, S) realizable if G can be drawn in the plane, such that only the pairs of edges in $R \cup S$ intersect, and all the pairs of edges in S do intersect. It is easy to see that this variant can also be decided in NP, since we can guess the relationship of the pairs in R and then solve the resulting AT-graph realizability problem (using Corollary 4.3.2).

Theorem 4.4.2 *The realizability of a topological expression can be decided in NP.*

Proof :

Given a topological expression φ over variables $(A_i)_{i \in I}$ we have to decide whether it can be realized by regions in the plane. We begin by simplifying φ, and then we show how to reduce the problem to a realizability problem (G, R, S) which we know to be in NP by the remarks preceding the Theorem.

We can guess a satisfying assignment of φ and all pairwise relationships between regions realizing φ. It is easy to check that the assignment is consistent with the relationships. We only need to verify that the given relationships are realizable. Hence we only need to consider explicit, conjunctive topological expression without negations. We can easily remove occurrences of equal, cover and contains.

Suppose that a topological graph (G, R, S) satisfies:

($\diamondsuit$) There are vertices z, z_1, z_2, z_3 in G connected to each other by edges which may not intersect any other edges.

- For each region R_i there is a vertex c_i (center) and a circle graph B_i (boundary) with at least 3 vertices, and no two edges of B_i may intersect.

(♣) Each vertex in B_i is connected to c_i, z_1, z_2, z_3; these edges are not allowed to intersect the boundary B_i, and no edge with endpoint c_i may intersect an edge with endpoint z_1, z_2, or z_3.

(▽) The boundaries B_i, B_j may share vertices unless disjoint(R_i, R_j), or inside(R_i, R_j) is contained in φ.

(△) Edges of B_i, B_j may intersect only if φ contains overlap(R_i, R_j).

- We say that a vertex v is an *in-R_i-witness* (*out-R_i-witness*) if it does not belong to B_i and is adjacent to c_i (z_1, z_2, and z_3, rsp). using an edge (edges, rsp). which are not allowed to intersect B_i.

(♠) If φ contains meet(R_i, R_j) or cover(R_i, R_j) then B_i and B_j share at least one common vertex.

- If disjoint(R_i, R_j) is in φ, then there is an out-R_i-witness on B_j, and an out-R_j-witness on B_i. If inside(R_i, R_j) then there is in-R_j-witness on B_i. If meet(R_i, R_j), then there is an out-R_i-witness on B_j between any two vertices shared with B_i, and an out-R_j-witness on B_i between any two vertices shared with B_j. If covered(R_i, R_j) then there is an in-R_i-witness on the boundary of B_j between any two vertices shared with B_i. If overlap(R_i, R_j) then there is an in-R_i witness and an out-R_i witness on the boundary of R_j, and vice versa.

We claim that if (G, R, S) has a weak realization then φ can be realized as an Euler diagram. Take the weak realization of (G, R, S). We can assume that z lies outside the triangle z_1, z_2, z_3. Hence by (◇) all other vertices and edges lie inside the triangle. Because of (♣) vertex c_i must lie inside of B_i (z_1, z_2, and z_3 being outside). Let region R_i be the interior of B_i together with its boundary. Clearly any in-R_i-witness lies inside R_i, and any out-R_i-witness lies outside R_i. For inside(R_i, R_j), *and* disjoint(R_i, R_j) boundaries cannot intersect and therefore the in/out-witnesses guarantee the correct relationship. For overlap(R_i, R_j) we have in/out-witnesses of overlap. For meet(R_i, R_j) the interior of R_i cannot intersect R_j, and vice versa because of the out-witnesses; similarly for covered(R_i, R_j).

Now we will show that if φ has an Euler diagram then there is (G, R, S) satisfying the above conditions which is small. This implies that the problem is in NP, because we can guess (G, R, S).

First redraw the graph using Lemma 4.4.1 so that the number of contact points is at most $|I|^2$. Enclose the diagram with a large region Z, on ∂Z choose three points z_1, z_2, z_3, choose z outside Z and connect z to z_1, z_2, z_3 with edges outside Z. Choose c_i inside each R_i. Now we will choose vertices on each ∂R_i and connect them to z_1, z_2, z_3 with edges inside $Z - R_i$ and to c_i with edges inside R_i (thus ($\clubsuit$) is satisfied). Clearly ($\triangledown$) is satisfied. All contact points will be chosen on each ∂R_i. This satisfies ($\spadesuit$) and also ($\triangle$), because we know that if two edges intersect then they intersect in an intersection point of their boudaries. If less than 3 points were chosen on ∂R_i, choose some more. Now it is routine check to see that we can choose in/out witnesses for disjoint(R_i, R_j), inside(R_i, R_j), meet(R_i, R_j), covered(R_i, R_j), and overlap(R_i, R_j). Note that we chose at most $|I|^2$ witnesses and at most $|I|^4$ in/out witnesses. Hence (G, R, S) is small and we can guess it in NP. ■

CHAPTER 5
CROSSING NUMBERS

5.1 Introduction

5.1.1 Crossing numbers

Let G be a graph with n vertices. Let M be a plane with n holes where each hole is labeled by a distinct vertex of G (i. e. we have a bijection between $V(G)$ and the holes of M). A **plane drawing** D of G is a collection of simple arcs $D = \{\gamma_{ij} \mid \{i,j\} \in E(G)\}$ such that γ_{ij} connects holes i and j on M, and no two arcs share an endpoint.

The **crossing number** $\operatorname{cr}(G)$ of a graph G is the minimum number of crossings in a plane drawing of G. More formally

$$\operatorname{cr}(G) = \min_D \sum_{\{e,f\}\in\binom{E}{2}} i(\gamma_e, \gamma_f), \tag{5.1}$$

where $i(\alpha, \beta)$ is the geometric intersection number of α and β. To see that this is equivalent to the standard definition of the crossing number note that geodesics intersect minimally (or see Lemma 2.3.8).

A **straight-line drawing** of a graph G is an assignment of vertices of G to the plane. Edge ij is drawn as a segment connecting the vertices i and j. The **rectilinear crossing number** $\operatorname{rcr}(G)$ of a graph G is the minimum number of pairs of crossing edges in a straight-line plane drawing of G. Clearly $\operatorname{rcr}(G) \geq \operatorname{cr}(G)$ for any graph G. A planar graph always has a planar straight-line embedding:

Theorem 5.1.1 ([SR34, Wag36, Fár48, Ste51]) $\operatorname{cr}(G) = 0 \;\Rightarrow\; \operatorname{rcr}(G) = 0.$

A natural question is whether $\mathrm{cr}(G) = \mathrm{rcr}(G)$ for all G. It turns out [Guy69] that already K_8 is a counterexample, $\mathrm{rcr}(K_8) = 19 > 18 = \mathrm{cr}(K_8)$. The gap between cr and rcr can be arbitrarily large:

Theorem 5.1.2 ([BD93]) *For any $k \geq 4$ there exists a graph G with $\mathrm{cr}(G) = 4$ and $\mathrm{rcr}(G) \geq k$.*

The **algebraic crossing number** $\mathrm{acr}(G)$, introduced by Tutte [Tut70], is the minimum sum of absolute values of algebraic intersection numbers of pairs of edges in a plane drawing of G, i. e.,

$$\mathrm{acr}(G) = \min_D \sum_{\{e,f\} \in \binom{E}{2}} |\hat{i}(\gamma_e, \gamma_f)|,$$

where $\hat{i}(\alpha, \beta)$ is the algebraic intersection number of α, β. Tutte was interested in a variant of the algebraic crossing number now called $\mathrm{acr}_-(G)$ (an exact definition is below) and asked whether $\mathrm{acr}_-(G) = \mathrm{cr}(G)$ for every graph G. Tutte's question is still open.

Pach and Tóth [PT00b] introduced the following two versions of crossing numbers. The **pair crossing number** $\mathrm{pcr}(G)$ is the minimum number of *pairs of edges that cross* in a drawing of G, i. e.,

$$\mathrm{pcr}(G) = \min_D \sum_{\{e,f\} \in \binom{E}{2}} [i(\gamma_e, \gamma_f) \neq 0],$$

where [cond] is 1 if cond is true and 0 otherwise. The **odd crossing number** $\mathrm{ocr}(G)$ is the minimum number of pairs of edges that cross odd number of times in a drawing of G, i. e.,

$$\mathrm{ocr}(G) = \min_D \sum_{\{e,f\} \in \binom{E}{2}} [i(\gamma_e, \gamma_f) \not\equiv 0 \pmod 2].$$

The following inequalities follow immediately from the definitions:

$$\mathrm{ocr}(G) \leq \mathrm{pcr}(G) \leq \mathrm{cr}(G) \leq \mathrm{rcr}(G), \text{ and}$$

$$\mathrm{ocr}(G) \leq \mathrm{acr}(G) \leq \mathrm{cr}(G) \leq \mathrm{rcr}(G).$$

No example of a graph G which contradicts $\mathrm{acr}(G) = \mathrm{pcr}(G) = \mathrm{cr}(G)$ is known.

If a graph which has an embedding in which all pairs of edges cross even number of times then the graph is planar:

Theorem 5.1.3 ([Han34, Tut70]) $\mathrm{ocr}(G) = 0 \ \Rightarrow\ \mathrm{cr}(G) = 0$.

This raised the question whether $\mathrm{cr}(G) = \mathrm{ocr}(G)$ for all graphs (problem #8 in [PT00a], [Pac00]). The main result of this chapter is:

Theorem 5.1.4 ([PSŠ05]) *For any $\varepsilon > 0$ there exists a graph G such that*

$$\mathrm{ocr}(G) < (\sqrt{3}/2 + \varepsilon)\mathrm{cr}(G).$$

In contrast to the arbitrary separation of $\mathrm{rcr}(G)$ and $\mathrm{cr}(G)$ (Theorem 5.1.2), the crossing number is bounded by a function of odd crossing number:

Theorem 5.1.5 ([PT00b]) *Let G be a graph. Then* $\mathrm{cr}(G) \leq 2(\mathrm{ocr}(G))^2$.

In terms of $\mathrm{pcr}(G)$ better upper bounds on $\mathrm{cr}(G)$ are known:

Theorem 5.1.6 ([Val04]) *Let G be a graph and let $k = \mathrm{pcr}(G)$. Then* $\mathrm{cr}(G) = O(k^2/\log k)$.

Theorem 5.1.7 ([KM04]) *Let G be a graph. Let $S = \sum_i d_i^2$ be the sum of squares of the degrees of vertices of G. Then*

$$\mathrm{cr}(G) = O\left((\mathrm{pcr}(G) + S)\log^3 n\right)$$

Following two variations on the definition of crossing numbers are given in [PT00a]: $\mathrm{cr}_+(G)$ in which the incident edges (i. e., edges which share a vertex) are not allowed to cross, and $\mathrm{cr}_-(G)$ in which the intersections of incident pairs of edges are ignored (i. e., the sum in (5.1) runs over pairs of non-incident edges). The $+, -$ variants are defined for $\mathrm{acr}, \mathrm{pcr}$, and ocr analogously. No example is known for which the

+, − variants differ. The variant $\mathrm{ocr}_-(G)$ is also called **independent odd crossing number** [Szé04]. Tutte [Tut70], in fact, proved a stronger result than Theorem 5.1.3; he showed that if $\mathrm{ocr}_-(G) = 0$ then $\mathrm{cr}(G) = 0$.

5.1.2 Complexity of computing crossing numbers

Let Cr be the decision version of the crossing number problem:

INSTANCE: graph G and an integer k.

QUESTION: Is $\mathrm{cr}(G) \leq k$?

Similarly let Pcr, Ocr, and Acr be the decision versions of the other crossing number problems. Let n be the number of vertices of G.

Cr is NP-complete [GJ83], even for cubic graphs [Hli04]. On the positive side Cr is fixed parameter tractable [Gro01]: there is an algorithm with running time $O(f(k)n^2)$. For bounded degree graphs there is $O(\log^2 n)$-approximation algorithm for $(n + \mathrm{cr}(G))$ ([EGS03], using [ARV04]). No non-trivial approximation algorithms are known if the degrees are not restricted. Computing the crossing number exactly seems to be difficult even for very regular graphs, e. g., we do not know the value of the crossing number of K_n for $n \geq 11$.

For the other crossing numbers we have: Rcr is NP-hard [Bie91], Pcr is NP-hard, and Ocr is NP-complete [PT00b]. We can show:

Theorem 5.1.8 ([SSŠ03]) Pcr *is* NP-*complete.*

5.2 Map crossing numbers

We will consider a variant of the crossing numbers in which the points on the holes are fixed. A **map** R on a surface M is a set $P = \{(a_1, b_1), \ldots, (a_m, b_m)\}$ of pairs of distinct points on ∂M together with positive weights $w_1, \ldots, w_m$. We require that a_i and b_i lie on different components of ∂M.

A **drawing** D of the map R is a set of m simple arcs $\gamma_1, \ldots, \gamma_m$ in S where γ_i connects a_i and b_i. Let

$$\operatorname{cr}(D) = \sum_{i<j} i(\gamma_i, \gamma_j) w_i w_j,$$

and $\operatorname{cr}(R) = \min_D \operatorname{cr}(D)$. We define the $\operatorname{pcr}(R)$, $\operatorname{acr}(R)$, and $\operatorname{ocr}(R)$ analogously.

Remark 1 For every map R,

$$\operatorname{ocr}(R) \leq \operatorname{pcr}(R) \leq \operatorname{cr}(R), \text{ and}$$

$$\operatorname{ocr}(R) \leq \operatorname{acr}(R) \leq \operatorname{cr}(R).$$

There is no example known which would contradict $\operatorname{cr}(R) = \operatorname{pcr}(R) = \operatorname{acr}(R)$. We can, however, separate $\operatorname{ocr}(R)$ from both $\operatorname{pcr}(R)$ and $\operatorname{acr}(R)$ for maps on the annulus.

Lemma 5.2.1 *The crossing numbers of maps* $\operatorname{cr}(R), \operatorname{pcr}(R), \operatorname{ocr}(R), \operatorname{acr}(R)$ *viewed as functions of* $w_1, \ldots, w_n$ *are continuous.*

Proof :

Consider the set V of vectors $v(\gamma_1, \ldots, \gamma_n) = (i(\gamma_i, \gamma_j))_{1 \leq i < j \leq n}$ for all possible curves $\gamma_1, \ldots, \gamma_n$. Let V' be the subset of vectors in V which do not dominate any other vector in V. Note that V' is finite and hence $\operatorname{cr}(R)$ is continuous in the w_i (it is a minimum of finitely many continuous functions). The same argument applies to the other crossing numbers. ■

Given a point a on ∂M, let $a + \varepsilon$ be a point in a small (i. e., one not containing any other point) neighborhood to the left of a. Similarly $a - \varepsilon$ is a point in a small neighborhood to the right of a.

Lemma 5.2.2 *Let R be a map on an oriented surface M. Let R' be a map obtained by replacing (a_i, b_i) by two pairs $(a - \varepsilon, b + \varepsilon)$ and $(a + \varepsilon, b - \varepsilon)$ with weights w_-, w_+ such that $w_- + w_+ = w_i$. Then $\operatorname{cr}(R') = \operatorname{cr}(R)$. The same is true for the other crossing numbers.*

Proof :

Given an optimal drawing for R we can draw both γ_+ and γ_- in a small neighborhood of γ_i. This shows $\mathrm{cr}(R') \le \mathrm{cr}(R)$. Given an optimal drawing D' for R' drawing γ_i in either a small neighborhood of γ_-, or γ_- gives $\mathrm{cr}(R) \le \mathrm{cr}(R')$. The same argument applies to the other crossing numbers. ■

5.2.1 Map crossing numbers on annulus

Let M be an annulus. We will call the two components of ∂M outer boundary and inner boundary. Let α be a non-trivial simple closed curve in M. Let a be on the outer boundary and b on the inner boundary of M. There is a correspondence between integers and isotopy (rel boundary) classes of curves with $\partial\gamma = \{a, b\}$. Fix a curve γ with $\partial\gamma = \{a, b\}$. Then $D_\alpha^k(\gamma),\ k \in \mathbb{Z}$ are representatives of these classes. For arcs γ, β connecting the outer and inner boundary:

$$\hat{\imath}(D_\alpha^k(\gamma), D_\alpha^\ell(\beta)) = k - \ell + \hat{\imath}(\gamma, \beta). \tag{5.2}$$

Note that $i(\gamma, \beta) = |\hat{\imath}(\gamma, \beta)|$ for any two curves γ, β on the annulus.

Let π be a permutation of $[n]$. A map R_π corresponding to π is constructed as follows. Choose $n + 1$ points on each component of ∂M and number them $0, 1, \ldots, n$ in the clockwise order. Let a_i be the vertex numbered i on the outer boundary and b_i be the vertex numbered π_i on the inner boundary, $i = 1, \ldots, n$. We ask a_i to be connected to b_i in R_π.

We will encode a drawing D of R_π by a sequence of n integers $x_1, \ldots, x_n$ as follows. Fix a curve β connecting the zeros and let γ_i be such that $i(\beta, \gamma_i) = 0$. We will connect a_i, b_i with arc $D_\alpha^{x_i}(\gamma_i)$ in D. Note that for $i < j$, $\hat{\imath}(\gamma_i, \gamma_j) = [\pi_i > \pi_j]$ and hence

$$\hat{\imath}(D_\alpha^{x_i}(\gamma_i), D_\alpha^{x_j}(\gamma_j)) = x_i - x_j + [\pi_i > \pi_j].$$

We have

$$\mathrm{acr}(R_\pi) = \mathrm{cr}(R_\pi) = \min\Big\{ \sum_{i<j} \big|x_i - x_j + [\pi_i > \pi_j]\big| w_i w_j \,\Big|\, x_i \in \mathbb{Z}, i \in [n] \Big\}, \tag{5.3}$$

$$\mathrm{pcr}(R_\pi) = \min\Big\{ \sum_{i<j} [x_i - x_j + [\pi_i > \pi_j] \neq 0] w_i w_j \,\Big|\, x_i \in \mathbb{Z}, i \in [n] \Big\}, \tag{5.4}$$

$$\mathrm{ocr}(R_\pi) = \min\Big\{ \sum_{i<j} [x_i - x_j + [\pi_i > \pi_j] \not\equiv 0 \pmod 2] w_i w_j \,\Big|\, x_i \in \mathbb{Z}, i \in [n] \Big\}. \tag{5.5}$$

Consider the relaxation of the integer program for $\mathrm{cr}(R_\pi)$:

$$\mathrm{cr}'(R_\pi) = \min\Big\{ \sum_{i<j} \big|x_i - x_j + [\pi_i > \pi_j]\big| w_i w_j \,\Big|\, x_i \in \mathbb{R}, i \in [n] \Big\}. \tag{5.6}$$

Since (5.6) is a relaxation of (5.3), we have $\mathrm{cr}'(R_\pi) \leq \mathrm{cr}(R_\pi)$. The following lemma shows $\mathrm{cr}'(R_\pi) = \mathrm{cr}(R_\pi)$.

Lemma 5.2.3 *Let n be a positive integer. Let $b_{ij} \in \mathbb{Z}$ and let $a_{ij} \in \mathbb{R}$ be non-negative, $1 \leq i < j \leq n$. Then*

$$\min\Big\{ \sum_{i<j} a_{ij} \big|x_i - x_j + b_{ij}\big| \,\Big|\, x_i \in \mathbb{R}, i \in [n] \Big\}$$

has an optimal solution with $x_i \in \mathbb{Z}$, $i \in [n]$.

Proof :

Let $\overline{x}^*$ be an optimal solution which satisfies the maximum number of $x_i - x_j + b_{ij} = 0$, $1 \leq i < j \leq n$. W. l. o. g., we can assume $x_1^* = 0$. Let G be a graph on vertex set $[n]$ with vertices i, j being connected if $x_i^* - x_j^* + b_{ij} = 0$. Note that if i, j are connected by an edge and one of x_i^*, x_j^* is an integer then both x_i^* and x_j^* are integers. It is enough to show that G is connected.

Suppose that G is not connected. There exists non-empty $A \subsetneq V(G)$ such that there are no edges between A and $V(G) - A$. Let 1_A be the characteristic vector of the set A (i. e., $(1_A)_i = (i \in A)$). Let $f(\lambda)$ be the value of the objective function on $\overline{x} = \overline{x}^* + \lambda \cdot 1_A$. Let I be the interval on which the signs of the $x_i - x_j + b_{ij}$, $1 \leq i < j \leq n$ do not change. Note that I is not the entire line. Since $f(\lambda)$ is linear on I and an open neighborhood of 0 is in I we have that f is constant on I. Choosing

$x = x^* + \lambda 1_A$ for λ an endpoint of I gives an optimal solution satisfying more $x_i - x_j + b_{ij} = 0,\ 1 \le i < j \le n$, a contradiction. ■

Theorem 5.2.4 *Crossing number of maps on annulus can be computed in polynomial time.*

Proof :
Note that $\mathrm{cr}'(R_\pi)$ is computed by the following linear program L_π.

$$\begin{aligned} &\min \sum_{i<j} y_{ij} w_i w_j \\ &y_{ij} \geq x_i - x_j + [\pi_i > \pi_j], \quad 1 \le i < j \le n \\ &y_{ij} \geq -x_i + x_j - [\pi_i > \pi_j], \quad 1 \le i < j \le n. \end{aligned}$$

■

The dual L'_π of L_π is

$$\begin{aligned} &\max \sum_{i<j} Q_{ij} [\pi_i > \pi_j] w_i w_j \\ &|Q_{ij}| \le w_i w_j, \quad 1 \le i < j \le n \\ &Q^T = -Q \\ &Q\overline{1} = \overline{0} \end{aligned}$$

Question 6 *Let R be a map on annulus. Can* $\mathrm{ocr}(R)$ *be computed in polynomial time?*

Conjecture 3 *For any map R on annulus* $\mathrm{cr}(R) = \mathrm{pcr}(R)$.

5.2.2 Example of a map with $\mathrm{ocr}(R) < \mathrm{pcr}(R)$

We will show that the map R in Figure 5.1 gives an example with $\mathrm{ocr}(R) < \mathrm{pcr}(R)$.

Lemma 5.2.5 *Let R be a map on the annulus given by $\pi = [1, 4, 3, 2]$ and weights a, b, d, c (in this order), where $a \le b \le c \le d$ and $a + c \ge d$. Then* $\mathrm{cr}(R) = \mathrm{pcr}(R) = ac + bd$ *and* $\mathrm{ocr}(R) = bc + ad$.

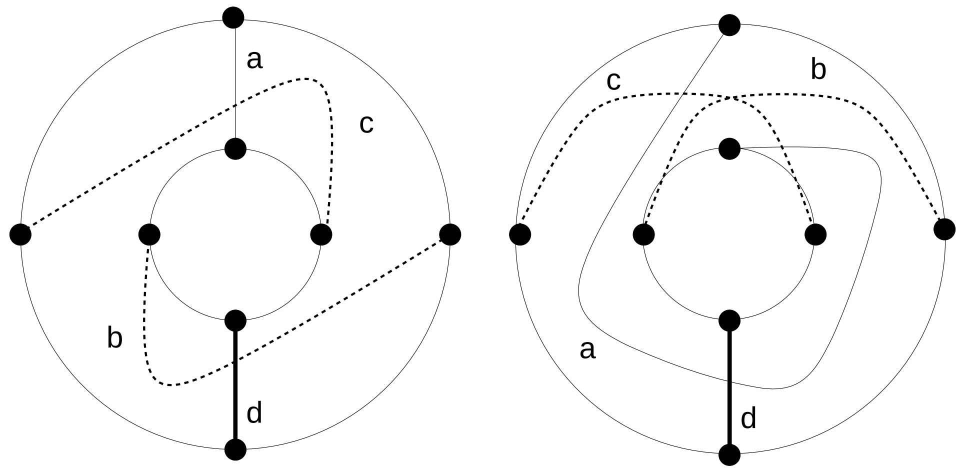

Figure 5.1: Weighed map R on the annulus such that $\text{ocr}(R) < \text{pcr}(R)$.

Remark 2 If the reader is interested only in the separation of $\text{ocr}(R)$ and $\text{cr}(R)$ then it is enough to verify the upper bound on ocr(R), and check that

$$Q = \begin{pmatrix} 0 & ab & a(c-b) & -ac \\ -ba & 0 & bd & b(a-d) \\ a(b-c) & -bd & 0 & b(d-a)+ac \\ ac & b(d-a) & b(a-d)-ac & 0 \end{pmatrix}$$

is a solution of the dual problem L'_π.

Proof of Lemma 5.2.5:

The upper bounds follow from the drawings in Figure 5.1, the left drawing for crossing, algebraic and pair crossing numbers, the right drawing for the odd crossing number. The lower bounds are tedious case analyses.

Claim: $\text{pcr}(R) \geq ac + bd$.

Proof of the claim: Let D be a drawing of R minimizing $\text{pcr}(D)$. We can apply twists so that the edge d is drawn as in the left part of Figure 5.1. Let α, β, γ be the number of clockwise twists that are applied to arcs a, b, c in the left part of Figure 5.1

to obtain the drawing D. Then,

$$\mathrm{pcr}(D) = cd[\gamma \neq 0] + bd[\beta \neq -1] + ad[\alpha \neq 0] + bc[\beta \neq \gamma] + ab[\alpha \neq \beta] + ac[\alpha \neq \gamma + 1]. \tag{5.7}$$

If $\gamma \neq 0$ then $\mathrm{pcr}(D) \geq cd + ab$ because at least one of the last five conditions in (5.7) must be true; the last five terms contribute at least ab (since $d \geq c \geq b \geq a$), and the first term contributes cd. Since $d(c-b) \geq a(c-b)$, $cd + ab \geq ac + bd$, and the claim is proved in the case that $\gamma \neq 0$.

Now assume that $\gamma = 0$. Equation (5.7) becomes

$$\mathrm{pcr}(D) = bd[\beta \neq -1] + bc[\beta \neq 0] + ad[\alpha \neq 0] + ac[\alpha \neq 1] + ab[\alpha \neq \beta]. \tag{5.8}$$

If $\beta \neq -1$ then $\mathrm{pcr}(D) \geq bd + ac$ because either $\alpha \neq 0$ or $\alpha \neq 1$. Since $bd + ac \geq bc + ad$, the claim is proved in the case that $\beta \neq -1$.

This leaves us with the case that $\beta = -1$. Equation (5.8) becomes

$$\mathrm{pcr}(D) = bc + ad[\alpha \neq 0] + ac[\alpha \neq 1] + ab[\alpha \neq -1]. \tag{5.9}$$

The right-hand side of Equation (5.9) is minimized for $\alpha = 0$. In this case $\mathrm{pcr}(D) = bc + ac + ab \geq ac + bd$ because we assume that $a + c \geq d$. □

Claim: $\mathrm{ocr}(R) \geq bc + ad$.

Proof of the claim: Let D be a drawing of R minimizing $\mathrm{ocr}(D)$. Let α, β, γ be as in the previous claim. We have

$$\mathrm{ocr}(D) = cd[\gamma]_2 + bd[\beta+1]_2 + ad[\alpha]_2 + bc[\beta+\gamma]_2 + ab[\alpha+\beta]_2 + ac[\alpha+\gamma+1]_2, \tag{5.10}$$

where $[x]_2$ is 0 if $x \equiv 0 \pmod 2$, and 1 otherwise.

If $\beta \not\equiv \gamma \pmod 2$ then the claim clearly follows unless $\gamma = 0$, $\beta = 1$, and $\alpha = 0$ (all modulo 2). In that case $\mathrm{ocr}(D) \geq bc + ab + ac \geq bc + ad$. Hence, the claim is proved if $\beta \not\equiv \gamma \pmod 2$.

Assume then that $\beta \equiv \gamma \pmod 2$. Equation (5.10) becomes

$$\mathrm{ocr}(D) = cd[\beta]_2 + bd[\beta+1]_2 + ad[\alpha]_2 + ab[\alpha+\beta]_2 + ac[\alpha+\beta+1]_2. \tag{5.11}$$

If $\alpha \equiv 1 \pmod 2$ then the claim clearly follows because either cd or bd contributes to the ocr. Thus we can assume $\alpha \equiv 0 \pmod 2$. Equation (5.11) becomes

$$\mathrm{ocr}(D) = (cd+ab)[\beta]_2 + (bd+ac)[\beta+1]_2. \tag{5.12}$$

For both $\beta \equiv 0 \pmod 2$ and $\beta \equiv 1 \pmod 2$ we get $\mathrm{ocr}(D) \geq bc + ad$. □ ■

Using $b = c = 1$, $a = (\sqrt{3}-1)/2$, and $d = 1+a$ we obtain the following separation.

Corollary 5.2.6 *There exists a map M on the annulus with* $\mathrm{ocr}(M)/\mathrm{pcr}(M) = \frac{\sqrt{3}}{2}$.

Conjecture 4 *For every map M on the annulus,* $\mathrm{ocr}(M) \geq \frac{\sqrt{3}}{2}\mathrm{cr}(M)$.

We can prove a weaker version of the conjecture 4 where $\sqrt{3}/2$ is replaced by $1/3$.

Lemma 5.2.7 *For every map R on the annulus*

$$\mathrm{cr}(R) \leq 3\mathrm{ocr}(R).$$

Proof :

We first consider the special case of unit weights. Let R consist of pairs of points $(a_1, b_1), \ldots, (a_n, b_n)$ with weights $w_1 = w_2 \ldots = w_n = 1$. Define the function $\mathrm{odd}(i,j,k)$ to be the odd crossing number of the map consisting of just three pairs: $(a_i, b_i), (a_j, b_j)$, and (a_k, b_k). Note that $\mathrm{odd}(i,j,k)$ is either 0 or 1.

Let D be a drawing of R minimizing $\mathrm{ocr}(D)$. If two curves γ_r and γ_s intersect an odd number of times in D then their contribution to $\mathrm{ocr}(D)$ is 1. Now,

$$\mathrm{ocr}(D) = \sum_{r<s} [i(\gamma_r, \gamma_s) \equiv 1 \pmod 2] = \frac{1}{n-2} \sum_{r<s, t \notin \{r,s\}} [i(\gamma_r, \gamma_s) \equiv 1 \pmod 2]. \tag{5.13}$$

By definition, $\text{odd}(i,j,k) = 1$ implies that $i(\gamma_r, \gamma_s) \equiv 1 \pmod 2$ for at least one pair $r, s \in \{i, j, k\}$. Therefore,

$$\sum_{i<j<k} \text{odd}(i,j,k) \le \sum_{r<s, t\notin\{r,s\}} [i(\gamma_r,\gamma_s) \equiv 1 \pmod 2]. \tag{5.14}$$

Let us look at the problem differently. Consider the drawing D_k in which γ_k is not crossed by any other curve. Obviously, $\text{cr}(R) \le \text{cr}(D_k)$. We have

$$\text{cr}(D_k) = \sum_{i<j \text{ s.t. } k\notin\{i,j\}} \text{odd}(i,j,k),$$

since three arcs can be drawn without crossings if their odd crossing number is zero. Hence,

$$\text{cr}(R) \le \frac{1}{n}\sum_k \text{cr}(D_k) \le \frac{1}{n} \sum_{i<j \text{ s.t. } k\notin\{i,j\}} \text{odd}(i,j,k) \le \frac{1}{n}\cdot 3 \cdot \sum_{i<j<k} \text{odd}(i,j,k). \tag{5.15}$$

Combining with (5.14) and (5.13) we obtain

$$\text{cr}(R) \quad \le \quad \frac{3(n-2)}{n}\text{ocr}(R) \le 3\text{ocr}(R).$$

By Lemma 5.2.2 the result extends to maps with integral weights. Scaling extends it to rational weights, and continuity (Lemma 5.2.1) to real weights. ■

5.3 Example of a graph with $\text{ocr}(G) < \text{pcr}(G)$

We modify the map R from Lemma 5.2.5 to obtain a graph G separating $\text{ocr}(G)$ and $\text{pcr}(G)$. The graph G will have integral weights on edges. From G we can get an unweighted graph G' with $\text{ocr}(G') = \text{ocr}(G)$ and $\text{pcr}(G') = \text{pcr}(G)$ by replacing an edge of weight w by w parallel edges of weight 1 (this does not change any of the crossing numbers). If needed we can get rid of parallel edges by the subdivision operation which does not change any of the crossing numbers.

We start with the map R from Lemma 5.2.5 with following integral weights:

$$a = \left\lfloor \frac{\sqrt{3}-1}{2} m \right\rfloor, \; b = c = m, \; d = \left\lfloor \frac{\sqrt{3}+1}{2} m \right\rfloor,$$

where $m \in \mathbb{Z}$ will be chosen later.

We replace each pair (a_i, b_i) in the map by w_i pairs $(a_{i,1}, b_{i,1}), \ldots, (a_{i,w_i}, b_{i,w_i})$ where the $a_{i,j}$ $(b_{i,j})$ occur on ∂M in clockwise order in a small interval around of a_i (b_i). By Lemma 5.2.2 the resulting map R_1 with unit weights will have the same crossing numbers as R. We then replace the boundaries of the annulus by cycles and obtain a graph G. We give weight $W = 1 + \mathrm{cr}(R)$ to the edges in the cycles. This ensures that in a drawing of G minimizing any of the crossing numbers the boundary cycles are disjointly embedded without self-intersections and that no edge intersects the boundary cycles.

Given a map R on annulus, the **flipped map** R' is obtained by flipping the order of points on one of the boundaries. There are two different ways of disjointly embedding the 2 boundary cycles of G on the sphere (depending on their relative orientations). In one of the cases the drawing D of G gives a drawing of R_1, in the other case it gives a drawing of the flipped map R_1'.

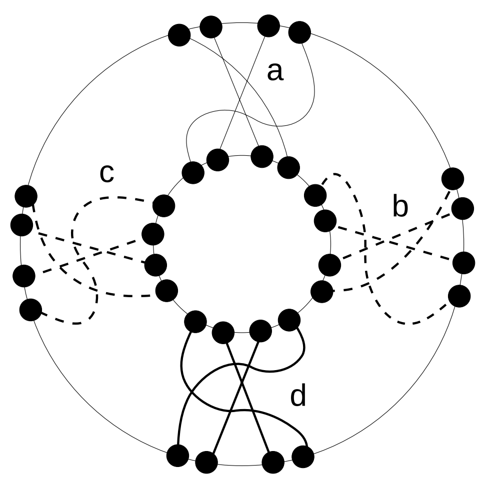

Figure 5.2: The flipped map R_1' resulting from opposite embedding of the boundary cycles of G.

Lemma 5.3.1 $\operatorname{ocr}(R_1') \geq 2m^2 - 4m.$

We will need the following result, which we prove later.

Lemma 5.3.2 *Let* $0 \leq a_1 \leq a_2 \leq \cdots \leq a_n$ *be such that* $a_n \leq a_1 + \cdots + a_{n-1}$. *Then*

$$\max_{|y_i| \leq a_i} \left(\left(\sum_{i=1}^{n} y_i \right)^2 - 2 \sum_{i=1}^{n} y_i^2 \right) = \left(\sum_{i=1}^{n} a_i \right)^2 - 2 \sum_{i=1}^{n} a_i^2.$$

Proof of Lemma 5.3.1:

As before let $w_1 = a, w_2 = b, w_3 = d, w_4 = c$. In any drawing of R_1' each "bunch" the edges split into two classes, those with even number of twists and those with odd number of twists. We can estimate (5.5) as follows.

$$\begin{aligned}
\operatorname{ocr}(R') = \min_{0 \leq k_i \leq w_i} \left(\sum_{i=1}^{4} \binom{k_i}{2} + \sum_{i=1}^{4} \binom{w_i - k_i}{2} + \sum_{i \neq j} k_i (w_j - k_j) \right) \geq \\
-\frac{1}{2} \sum_{i=1}^{4} w_i + \min_{0 \leq x_i \leq w_i} \left(\sum_{i=1}^{4} \frac{x_i^2}{2} + \sum_{i=1}^{4} \frac{(w_i - x_i)^2}{2} + \sum_{i \neq j} x_i (w_j - x_j) \right) = \\
-\frac{1}{2} \sum_{i=1}^{4} w_i + \frac{1}{4} \left(\sum_{i=1}^{4} w_i \right)^2 + \min_{|y_i| \leq w_i/2} \left(2 \sum_{i=1}^{4} y_i^2 - \left(\sum_{i=1}^{4} y_i \right)^2 \right) \geq \\
\frac{1}{2} \sum_{i=1}^{4} w_i^2 - \frac{1}{2} \sum_{i=1}^{4} w_i \geq \\
\frac{1}{2} \left(\left(\frac{\sqrt{3}+1}{2} m - 1 \right)^2 + 2m^2 + \left(\frac{\sqrt{3}-1}{2} m - 1 \right)^2 - 4m \right) \geq 2m^2 - 4m.
\end{aligned} \tag{5.16}$$

■

Proof of Theorem 5.1.4

For the graph G, by Lemma 5.2.5, we have

$$\operatorname{ocr}(G) \leq \operatorname{ocr}(R) = w_1 w_3 + w_2 w_4 \leq \frac{3}{2} m^2,$$

and by Lemmas 5.2.5, and 5.3.1, we have

$$\mathrm{cr}(G) \geq \min\{\mathrm{cr}(R), \mathrm{ocr}(R')\} \geq \min\left\{\sqrt{3}m^2 - 2m, 2m^2 - 4m\right\}.$$

For any $\varepsilon > 0$ there exists m such that $\mathrm{ocr}(G) < (\sqrt{3}/2 + \varepsilon)\mathrm{cr}(G)$. ■

Proof of Lemma 5.3.2:

Let $y_1, \ldots, y_n$ achieve the maximum value. Replacing the y_i by $|y_i|$ does not decrease the objective function. W. l. o. g., we can assume $0 \leq y_1 \leq y_2 \leq \cdots \leq y_n$. Note that $y_i < y_j$ then $y_i = a_i$ (otherwise increasing y_i by ε and decreasing y_j by ε increases the objective function for small ε).

Let k be the largest i such that $y_i = a_i$. Let $k = 0$ if no such i exists. We have $y_i = a_i$ for $i \leq k$ and $y_{k+1} = \cdots = y_n$. If $k = n$ we are done. Let

$$f(t) = \left(\sum_{i=1}^{k} a_i + (n-k)t\right)^2 - 2\left(\sum_{i=1}^{k} a_i^2 + (n-k)t^2\right).$$

We have

$$f'(t) = 2(n-k)\left(\sum_{i=1}^{k} a_i + (n-k-2)t\right).$$

Note that for $t < a_{k+1}$ we have $f'(t) > 0$ and hence the only optimal choice is $t = a_{k+1}$. Hence $y_{k+1} = a_{k+1}$, a contradiction with our choice of k. ■

REFERENCES

[AAFG01] Iris Anshel, Michael Anshel, Benji Fisher, and Dorian Goldfeld. New key agreement protocols in braid group cryptography. In *Topics in cryptology—CT-RSA 2001 (San Francisco, CA)*, volume 2020 of *Lecture Notes in Comput. Sci.*, pages 13–27. Springer, Berlin, 2001.

[AH77] Kenneth Appel and Wolfgang Haken. Every planar map is 4-colorable. *Illinois Journal of Mathematics*, 21:429–567, 1977.

[AM90] Selman Akbulut and John D. McCarthy. *Casson's invariant for oriented homology* 3*-spheres*, volume 36 of *Mathematical Notes*. Princeton University Press, Princeton, NJ, 1990. An exposition.

[ARV04] Sanjeev Arora, Satish Rao, and Umesh V. Vazirani. Expander flows, geometric embeddings and graph partitioning. In *STOC*, pages 222–231, 2004.

[BD93] Daniel Bienstock and Nathaniel Dean. Bounds for rectilinear crossing numbers. *J. Graph Theory*, 17(3):333–348, 1993.

[Ben59] Seymour Benzer. On the topology of the genetic fine structure. *Proceedings of the National Academy of Science*, 45:1607–1620, 1959.

[Bie91] Daniel Bienstock. Some provably hard crossing number problems. *Discrete Comput. Geom.*, 6(5):443–459, 1991.

[BS84] Joan S. Birman and Caroline Series. An algorithm for simple curves on surfaces. *J. London Math. Soc. (2)*, 29(2):331–342, 1984.

[BS87] Joan S. Birman and Caroline Series. Dehn's algorithm revisited, with applications to simple curves on surfaces. In *Combinatorial group theory and topology (Alta, Utah, 1984)*, volume 111 of *Ann. of Math. Stud.*, pages 451–478. Princeton Univ. Press, Princeton, NJ, 1987.

[CGP98] Zhi-Zhong Chen, Michelangelo Grigni, and Christos Papadimitriou. Planar map graphs. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC-98)*, pages 514–523, New York, 1998. ACM Press.

[Chi69] D. R. J. Chillingworth. Simple closed curves on surfaces. *Bull. London Math. Soc.*, 1:310–314, 1969.

[Chi71] D. R. J. Chillingworth. An algorithm for families of disjoint simple closed curves on surfaces. *Bull. London Math. Soc.*, 3:23–26, 1971.

[CL87] Marshall Cohen and Martin Lustig. Paths of geodesics and geometric intersection numbers. I. In *Combinatorial group theory and topology (Alta, Utah, 1984)*, volume 111 of *Ann. of Math. Stud.*, pages 479–500. Princeton Univ. Press, Princeton, NJ, 1987.

[CLL+02] Moses Charikar, Eric Lehman, Ding Liu, Rina Panigrahy, Manoj Prabhakaran, April Rasala, Amit Sahai, and Abhi Shelat. Approximating the smallest grammar: Kolmogorov complexity in natural models, 2002. In Proc. 29th Ann. Sympo. on Theory of Computing, 792-801.

[Dey94] Tamal Dey. A new technique to compute polygonal schema for 2-manifolds with application to null-homotopy detection. In *Symposium on Computational Geometry*, pages 277–284, 1994.

[DG99] Tamal Dey and Sumanta Guha. Transforming curves on surfaces. *JCSS: Journal of Computer and System Sciences*, 58(2):297–325, 1999.

[DK02] Volker Diekert and Manfred Kufleitner. A remark about quadratic trace equations. In *Developments in Language Theory*, pages 59–66, 2002.

[DMM97] Volker Diekert, Yuri Matiyasevich, and Anca Muscholl. Solving trace equations using lexicographical normal forms. In *Proceedings of the24th International Colloquium on Automata, Languages and Programming (ICALP'97), Bologna (Italy) 1997*, Berlin-Heidelberg-New York, 1997. Springer.

[DS95] 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.

[Ege91] Max J. Egenhofer. Reasoning about binary topological relations. *Lecture Notes in Computer Science*, 525:143–160, 1991.

[EGS03] Guy Even, Sudipto Guha, and Baruch Schieber. Improved approximations of crossings in graph drawings and VLSI layout areas. *SIAM J. Comput.*, 32(1):231–252 (electronic), 2002/03.

[Fár48] István Fáry. On the straight line representations of planar graphs. *Acta Sci. Math.*, 11:229–233, 1948.

[FLP79] A. Fathi, F. Laudenbach, and V. Poénaru. Travaux de Thurston sur les surfaces. *Astérisque*, 66–67, 1979.

[FM97] A. T. Fomenko and S. V. Matveev. *Algorithmic and computer methods for three-manifolds*, volume 425 of *Mathematics and its Applications*. Kluwer Academic Publishers, Dordrecht, 1997. Translated from the 1991 Russian original by M. Tsaplina and Michiel Hazewinkel and revised by the authors, With a preface by Hazewinkel.

[FT91] Benson Farb and William P. Thurston. Homeomorphisms and simple closed curves. manuscript, 1991.

[FT98] Martin Farach and Mikkel Thorup. String matching in lempel-ziv compressed strings. *Algorithmica*, 20(4):388–404, 1998.

[Ful95] William Fulton. *Algebraic topology*, volume 153 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1995. A first course.

[GJ83] Michael R. Garey and David S. Johnson. Crossing number is NP-complete. *SIAM Journal on Algebraic and Discrete Methods*, 4(3):312–316, 1983.

[GKPR96] Leszek Gąsieniec, Marek Karpinski, Wojciech Plandowski, and Wojciech Rytter. Randomized efficient algorithms for compressed strings: The finger-print approach (extended abstract). In *Proceedings of the 7th Annual Symposium on Combinatorial*, pages 39–49, 1996.

[GPP95] Michelangelo Grigni, Dimitris Papadias, and Christos Papadimitriou. Topological inference. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 901–906, 1995.

[Gra76] Ron L. Graham. Problem 1. In *Open Problems at 5th Hungarian Colloquium on Combinatorics*, 1976.

[Gre60] Martin Greendlinger. On Dehn's algorithms for the conjugacy and word problems, with applications. *Comm. Pure Appl. Math.*, 13:641–677, 1960.

[Gro01] Martin Grohe. Computing crossing numbers in quadratic time. In *Proceedings of the 32nd ACM Symposium on Theory of Computing*, pages 231–236, 2001.

[Guy69] Richard K. Guy. The decline and fall of Zarankiewicz's theorem. In *Proof Techniques in Graph Theory (Proc. Second Ann Arbor Graph Theory Conf., Ann Arbor, Mich., 1968)*, pages 63–69. Academic Press, New York, 1969.

[Hag98] Torben Hagerup. Sorting and searching on the word RAM. In *STACS 98 (Paris, 1998)*, volume 1373 of *Lecture Notes in Comput. Sci.*, pages 366–398. Springer, Berlin, 1998.

[Han34] Chaim Chojnacki (Haim Hanani). Über wesentlich unplättbare Kurven im drei-dimensionalen Raume. *Fundamenta Mathematicae*, 23:135–142, 1934.

[Hli04] Petr Hliněný. Crossing number is hard for cubic graphs. *ITI Technical report 176*, 2004.

[HT80] Allen Hatcher and William Thurston. A presentation for the mapping class group of a closed orientable surface. *Topology*, 19(3):221–237, 1980.

[HT97] Hessam Hamidi-Tehrani. *Algorithms in the Mapping Class Groups.* PhD thesis, Columbia, 1997.

[HTC96] Hessam Hamidi-Tehrani and Zong-He Chen. Surface diffeomorphisms via train-tracks. *Topology and its Applications*, 73:141–167, 1996.

[Imr84] Wilfried Imrich. Explicit construction of regular graphs without small cycles. *Combinatorica*, 4(1):53–59, 1984.

[KGK86] Ján Kratochvíl, Miroslav Goljan, and Petr Kučera. String graphs. *Rozpravy Československ Akad. Věd Řada Mat. Přírod. Věd*, 3(96), 1986.

[KLC+00] Ki Hyoung Ko, Sang Jin Lee, Jung Hee Cheon, Jae Woo Han, Ju-sung Kang, and Choonsik Park. New public-key cryptosystem using braid groups. In *Advances in cryptology—CRYPTO 2000 (Santa Barbara, CA)*, volume 1880 of *Lecture Notes in Comput. Sci.*, pages 166–183. Springer, Berlin, 2000.

[KM91] Ján Kratochvíl and Jiří Matoušek. String graphs requiring exponential representations. *Journal of Combinatorial Theory, Series B*, 53:1–4, 1991.

[KM04] Petr Kolman and Jiří Matoušek. Crossing number, pair-crossing number, and expansion. *J. Combin. Theory Ser. B*, 92(1):99–113, 2004.

[Kne30] H. Kneser. Geschlossene Flächen in dreidimensionalen Mannigfaltigkeiten. *Jahresbericht der Deutschen Mathematikver-Vereinigung*, pages 248–260, 1930.

[Kra91] Ján Kratochvíl. String graphs. II. recognizing string graphs is NP-hard. *Journal of Combinatorial Theory, Series B*, 52, 1991.

[Lee00] John M. Lee. *Introduction to topological manifolds*, volume 202 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2000.

[Lei84] Frank Thomson Leighton. New lower bound techniques for VLSI. *Math. Systems Theory*, 17(1):47–70, 1984.

[Lic64] W. B. R. Lickorish. A finite set of generators for the homeotopy group of a 2-manifold. *Proc. Cambridge Philos. Soc.*, 60:769–778, 1964.

[Lic66] W. B. R. Lickorish. Corrigendum: On the homeotopy group of a 2-manifold. *Proc. Cambridge Philos. Soc.*, 62:679–681, 1966.

[LS02] Eric Lehman and Abhi Shelat. Approximations algorithms for grammar-based compression, 2002. In Thirteenth Annual Symposium on Discrete Algorithms (SODA'02).

[Luo01] Feng Luo. Some applications of a multiplicative structure on simple loops in surfaces. In *Knots, braids, and mapping class groups—papers dedicated to Joan S. Birman (New York, 1998)*, volume 24 of *AMS/IP Stud. Adv. Math.*, pages 123–129. Amer. Math. Soc., Providence, RI, 2001.

[Lus87] Martin Lustig. Paths of geodesics and geometric intersection numbers. II. In *Combinatorial group theory and topology (Alta, Utah, 1984)*, volume 111 of *Ann. of Math. Stud.*, pages 501–543. Princeton Univ. Press, Princeton, NJ, 1987.

[LZ76] Abraham Lempel and Jacob Ziv. On the complexity of finite sequences. *IEEE Trans. Information Theory*, IT-22(1):75–81, 1976.

[Mak77] Gennady S. Makanin. The problem of the solvability of equations in a free semigroup. *Mat. Sb. (N.S.)*, 103(145)(2):147–236, 319, 1977.

[Mas91] William S. Massey. *A basic course in algebraic topology*, volume 127 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1991.

[MNT88] Jiří. Matoušek, Jaroslab Nešetřil, and Robin Thomas. On polynomial time decidability of induced-minor-closed classes. *Comment. Math. Univ. Carolin.*, 29(4):703–710, 1988.

[Moi52] Edwin E. Moise. Affine structures in 3-manifolds. V. The triangulation theorem and Hauptvermutung. *Ann. of Math. (2)*, 56:96–114, 1952.

[MST97] Masamichi Miyazaki, Ayumi Shinohara, and Masayuki Takeda. An improved pattern matching algorithm for strings in terms of straight-line programs. In A. Apostolico and J. Hein, editors, *Proceedings of the 8th Annual Symposium on Combinatorial Pattern Matching*, volume 1264, pages 1–11, Aarhus, Denmark, 1997. Springer-Verlag, Berlin.

[Pac00] János Pach. Crossing numbers. In *Discrete and computational geometry (Tokyo, 1998)*, volume 1763 of *Lecture Notes in Comput. Sci.*, pages 267–273. Springer, Berlin, 2000.

[Pen84] Robert C. Penner. The action of the mapping class group on curves in surfaces. *L'Enseignement Mathématique*, 30:39–55, 1984.

[Pla99] Wojciech Plandowski. Satisfiability of word equations with constants is in PSPACE. In *40th Annual Symposium on Foundations of Computer Science (New York, 1999)*, pages 495–500. IEEE Computer Soc., Los Alamitos, CA, 1999.

[Poi95] Henri Poincaré. Analysis situs. *J. Ecole Polytechn.*, 2(1):1–121, 1895.

[PR98] Wojciech Plandowski and Wojciech Rytter. Application of Lempel-Ziv encodings to the solution of words equations. In *Automata, Languages and Programming*, pages 731–742, 1998.

[PSŠ05] Michael Pelsmajer, Marcus Schaefer, and Daniel Štefankovič. Odd crossing number is not crossing number. manuscript, 2005.

[PT00a] János Pach and Géza Tóth. Thirteen problems on crossing numbers. *Geombinatorics*, 9(4):194–207, 2000.

[PT00b] János Pach and Géza Tóth. Which crossing number is it anyway? *Journal of Combinatorial Theory, Series B*, 80:225–246, 2000.

[PT01] János Pach and Géza Tóth. Recognizing string graphs is decidable. In P. Mutzel, editor, *Graph Drawing 2001*, Lecture Notes in Computer Science, pages 247–260. Springer, 2001.

[Rad25] Tibor Radó. Über den Begriff der Riemannschen Fläche. *Acta Litt. Sci. Szeged*, 2:101–121, 125.

[RD99] John M. Robson and Volker Diekert. On quadratic word equations. *Lecture Notes in Computer Science*, 1563:217–226, 1999.

[Rei62] Bruce L. Reinhart. Algorithms for Jordan curves on compact surfaces. *Ann. of Math. (2)*, 75:209–222, 1962.

[Ryt02] Wojciech Rytter. Application of Lempel-Ziv factorization to the approximation of grammar-based compression, 2002. In Proceedings 13th Annnual Symposium Combinatorial Pattern Matching, 20-31, 2002.

[Sch92] Haijo Schipper. Determining contractibility of curves. In *Symposium on Computational Geometry*, pages 358–367, 1992.

[Sin66] Frank W. Sinden. Topology of thin film circuits. *Bell System Technical Journal*, 45:1639–1662, 1966.

[SR34] Ernst Steinitz and Hans Rademacher. *Vorlesungen über die Theorie der Polyeder*. Springer, Berlin, 1934.

[SŠ00] Marcus Schaefer and Daniel Štefankovič. Decidability of string graphs. TR-000-05, 2000.

[SŠ01] Marcus Schaefer and Daniel Štefankovič. Decidability of string graphs. In *Proceedings of the 33th Annual ACM Symposium on Theory of Computing (STOC-2001)*, pages 241–246, 2001. Invited to the JCSS special issue on STOC'01.

[SŠ04] Marcus Schaefer and Daniel Štefankovič. Decidability of string graphs. *J. Comput. System Sci.*, 68(2):319–334, 2004.

[SSŠ02] Marcus Schaefer, Eric Sedgwick, and Daniel Štefankovič. Recognizing string graphs in NP. In *Proceedings of the 33th Annual ACM Symposium on Theory of Computing (STOC-2002)*, pages 1–6, 2002.

[SSŠ03] Marcus Schaefer, Eric Sedgwick, and Daniel Štefankovič. Recognizing string graphs in NP. *J. Comput. System Sci.*, 67(2):365–380, 2003. Special issue on STOC2002 (Montreal, QC).

[Ste51] S. K. Stein. Convex maps. *Proc. Amer. Math. Soc.*, 2:464–466, 1951.

[Sti93] John Stillwell. *Classical topology and combinatorial group theory*, volume 72 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, second edition, 1993.

[Sti99] John Stillwell. Max Dehn. In *History of topology*, pages 965–978. North-Holland, Amsterdam, 1999.

[Szé04] László A. Székely. A successful concept for measuring non-planarity of graphs: the crossing number. *Discrete Math.*, 276(1-3):331–352, 2004. 6th International Conference on Graph Theory.

[Tho98] Mikkel Thorup. Map graphs in polynomial time. In *IEEE Symposium on Foundations of Computer Science*, pages 396–405, 1998.

[Tri95] Claude Tricot. *Curves and Fractal Dimension*. Springer, 1995.

[Tut70] William T. Tutte. Toward a theory of crossing numbers. *J. Combinatorial Theory*, 8:45–53, 1970.

[Val04] Pavel Valtr. On the pair-crossing number. manuscript, 2004.

[Wag36] K. Wagner. Bemerkungen zum Vierfarbenproblem. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 46:26–32, 1936.

[Zie65] Heiner Zieschang. Algorithmen für einfache Kurven auf Flächen. *Math. Scand.*, 17:17–40, 1965.