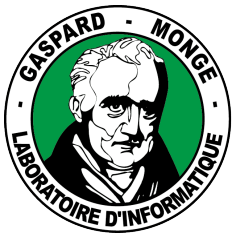

Sous la co-tutelle de :
CNRS
ÉCOLE DES PONTS PARISTECH
UNIVERSITÉ GUSTAVE EIFFEL

PARIS EST SUP

Mémoire d'Habilitation à Diriger des Recherches

Spécialité : Informatique

par

ARNAUD DE MESMAY

Interactions between algorithms, geometry and topology in low dimensions

Soutenu publiquement le 16 Février 2022 devant un jury composé de

Mr. Sergio CABELLO (University of Ljubljana)	Examinateur
Mr. Victor CHEPOI (Aix-Marseille Université)	Examinateur
Mr. Éric FUSY (CNRS, Université Gustave Eiffel)	Examinateur
Mr. Xavier GOAOC (Université de Lorraine)	Rapporteur
Mr. Joel HASS (University of California, Davis)	Rapporteur
Mme Claire MATHIEU (CNRS, Université de Paris)	Examinatrice
Mr. Hugo PARLIER (Université du Luxembourg)	Examinateur

Autre rapportrice :
Mme Yusu WANG (University of California, San Diego)

Unité mixte de recherche 8049 : Laboratoire d'Informatique Gaspard Monge
École doctorale 532 : MSTIC Université Paris Est

Résumé

Ce mémoire d'habilitation présente un panorama de mes activités de recherche depuis ma soutenance de thèse. Le thème commun à tous mes travaux est l'étude des interactions entre les propriétés algorithmiques, géométriques et topologiques d'objets mathématiques de basse dimension, tels que les graphes et leurs plongements sur les surfaces, ainsi que les nœuds et les 3-variétés.

Ce document présente principalement trois séries de travaux que j'ai menés avec de nombreux co-auteurs sur plusieurs années. La première traite de problèmes de décompositions de surfaces : quelle est la façon optimale de découper une surface en un disque ? Par dualité, ce problème est relié au problème algorithmique MULTICUT pour les graphes plongés, qui vise à calculer la meilleure façon de découper un graphe pour séparer un ensemble de paires de terminaux. Nous utilisons des algorithmes topologiques pour obtenir des schémas d'approximation pour ces deux problèmes, et parallèlement nous développons de nouvelles techniques pour prouver des bornes inférieures algorithmiques pour des problèmes sur des graphes plongés, en particulier ces deux-ci. Ensuite, nous nous intéressons au problème de calculer une homotopie optimale, c'est-à-dire où la longueur maximale des courbes est minimisée. L'étude de telles homotopies est motivée par des considérations à la fois mathématiques et plus appliquées. Nous établissons des propriétés structurelles fortes sur ces homotopies optimales, valables à la fois dans un contexte riemannien et discret, qui nous permettent d'obtenir des algorithmes pour les calculer, aussi bien de façon exacte qu'approchée. Nous établissons aussi des liens avec la théorie des mineurs de graphes. Dans un troisième temps, nous présentons une famille de résultats de difficulté algorithmique pour des problèmes en théorie des nœuds et en topologie de dimension trois. En particulier nous prouvons la NP-difficulté du problème de décider la plongeabilité de complexes simpliciaux dans $\mathbb{R}^3$, ainsi que du problème de démêler un nœud trivial en un nombre minimal de mouvements. Nous exposons ensuite de façon plus succincte d'autres contributions, sur les plongements par plus courts chemins de graphes sur les surfaces, le raccourcissement des courbes et les transformations électriques, la largeur d'arborescence des diagrammes de nœuds et le nombre de croisements d'entrelacs. Nous concluons avec quelques perspectives et directions de recherche ultérieures.

Abstract

This habilitation thesis presents an overview of my research activities since the defense of my PhD Thesis. The common theme underlying most of my work is the investigation of the interactions between the algorithmic, geometric and topological properties of a mathematical object. The focus is on low dimensions: graphs and their embeddings on surfaces, as well as knots and 3-dimensional topology.

This document showcases three lines of work that I have developed with multiple coauthors over the course of several years. The first one deals with a problem of surface decomposition: what is the best way to cut a surface into a disk? By duality, this problem is related to the algorithmic problem MULTICUT for embedded graphs, asking for the best way to cut a graph so as to separate a specified set of pairs of terminals. We leverage topological algorithms to provide approximation schemes for both problems, and dually we develop new tools to establish lower bounds for computational problems involving embedded graphs, including these two. We then turn our attention towards the problem of computing optimal homotopies, i.e., homotopies where the length of the longest curve is minimized. Their study is motivated by questions coming both from Riemannian and computational geometry. We prove strong results on the structure of optimal homotopies, valid in both the discrete and the continuous setting, and leverage those to provide improved algorithms to compute optimal homotopies exactly or approximatively, as well as to draw connections with graph minor theory. In a third step, we present a family of computational hardness results for various problems in knot theory and 3-dimensional topology. Most importantly we prove the NP-hardness for the problem of deciding embeddability of simplicial complexes in $\mathbb{R}^3$ and for the problem of optimally untangling a diagram of the unknot. We then survey more succinctly some other contributions pertaining to shortest path embeddings of graphs on surfaces, curve tightening and electrical transformations, treewidth of knot diagrams and the crossing number of links. We conclude with some perspectives and future research directions.

Table of Contents

Title Page i
Résumé ii
Abstract iii
Table of Contents v
Acknowledgments vii
Complete list of publications ix

1 Introduction **1**

2 Introduction en français **7**

3 Preliminaries **13**

4 Cutting a surface: approximation schemes and lower bounds **23**
4.1 Introduction 23
4.2 Approximation schemes for the Shortest Cut Graph and Multicut problems 28
4.3 Lower bounds 32

5 Homotopy height **41**
5.1 Introduction 41

5.2 Structure of optimal homotopies 46
5.3 Algorithmic implications 50
5.4 Grid-major height 52

6 Hard topological problems in three dimensions 57
6.1 Introduction 57
6.1.1 Hard problems in knot theory 57
6.1.2 Embeddability questions and their computational complexity 60
6.2 Borromean gadgets and hardness proofs 62
6.2.1 The trivial sublink problem 62
6.2.2 Unlinking number and 4-dimensional problems 63
6.2.3 Optimal number of Reidemeister moves 66
6.3 Hardness of embeddability into 3 dimensions 69

7 Other works on surfaces 75
7.1 Shortest path embeddings and crossing numbers 75
7.2 Homotopy moves 78

8 Other works on knots and 3-manifolds 83
8.1 Treewidth of knots 83
8.2 Hardness of the link crossing number 85

9 Perspectives 89
9.1 Work with students 89
9.2 Further research directions 91

Bibliography 95

Acknowledgments

I would first like to address my heartfelt thanks to Xavier Goaoc, Joel Hass and Yusu Wang, who agreed to review this manuscript and who all gave me precious feedback. I am also very grateful to Sergio Cabello, Victor Chepoi, Éric Fusy, Claire Mathieu and Hugo Parlier for agreeing to be in the examing committee. The sanitary conditions made it so that the defense had to happen online, and thus the traditional "pot" could not happen: I hope that I will find a different occasion to thank you all in person in the future.

I have been working with many coauthors, and while this document is mostly written in the first person, none of the works presented here could have existed without them. Many thanks to Vincent Cohen-Addad, Terese Biedl, Ben Burton, Erin W. Chambers, Gregory R. Chambers, Hsien-Chih Chang, Éric Colin de Verdière, David Eppstein, Jeff Erickson, Alfredo Hubard, Vojtěch Kaluža, David Letscher, Dàniel Marx, Tim Ophelders, Jessica Purcell, Yo'av Rieck, Marcus Schaefer, Saul Schleimer, Eric Sedgwick, Martin Tancer, Dylan Thurston, Stephan Tillmann and Uli Wagner. I would like to express a particular thank you to Éric Colin de Verdière, whose strong and steady support has been an incredible boon for me in the past ten years, to Francis Lazarus for welcoming me so warmly in Grenoble, and to Vincent Cohen-Addad and Alfredo Hubard for the long-time collaboration and friendship. Thanks also to my students Jean Chartier, Niloufar Fuladi, and Corentin Lunel, and to all the other people I have had the chance to talk to and work with but who are too numerous to be named.

I have benefited from a great work environment at IST Austria, then at Gipsa-lab and now at LIGM and would like to thank my colleagues in all these institutions for their everyday help and support, and in particular to Philippe Gambette for very helpful discussions regarding the habilitation.

Finally, my last thoughts go to my family, Olivier and Nelly of course, Frédéric and Aude of course as well, the bright little shining star Manon, and the indispensable and *incorreggibìle* Fanny.

Complete list of publications

Articles that I have co-authored are listed alphabetically, the other references are listed numerically. This list is ordered in reverse chronological order. Each article is only listed once, even when it appeared in multiple versions (for example in the proceedings of a conference and then in a journal). All of these articles are freely available on my webpage, and most of them are on arXiv. Co-authors who were students at the time of writing are starred.

Publications presented in this document

[A] H.-C. CHANG* AND A. DE MESMAY, *Tightening curves on surfaces monotonically with applications*, in Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2020, pp. 747–766. Cited on page 78, 80, 82, 89

[B] A. DE MESMAY, M. SCHAEFER, AND E. SEDGWICK, *Link crossing number is NP-hard*, Journal of Knot Theory and Its Ramifications, (2020), Volume 29, Number 6, p. 2050043. Cited on page 85, 86

[C] T. BIEDL, E. W. CHAMBERS, D. EPPSTEIN, A. DE MESMAY, AND T. OPHELDERS*, *Homotopy height, grid-major height and graph-drawing height*, in International Symposium on Graph Drawing and Network Visualization, Springer, 2019, pp. 468–481. Cited on page 41, 52, 53, 54, 55, 89

[D] V. COHEN-ADDAD, É. COLIN DE VERDIÈRE, D. MARX, AND A. DE MESMAY, *Almost tight lower bounds for hard cutting problems in embedded graphs*, Journal of the ACM, Volume 68, Issue 4, August 2021. An extended abstract appear in 35th International Symposium on Computational Geometry (SoCG 2019), Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019. Best paper award. Cited on page 23, 24, 33, 36, 37, 38, 91

[E] A. DE MESMAY, Y. RIECK, E. SEDGWICK, AND M. TANCER, *The unbearable hardness of unknotting*, Advances in Mathematics, Volume 381, 2021. An extended abstract

appeared in 35th International Symposium on Computational Geometry (SoCG 2019), Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019. Cited on page 57, 59, 60, 65, 66, 68

[F] A. DE MESMAY, J. PURCELL, S. SCHLEIMER, AND E. SEDGWICK, *On the tree-width of knot diagrams*, Journal of Computational Geometry, 10 (2019), pp. 164–180. Cited on page 83, 84, 90

[G] E. W. CHAMBERS, G. R. CHAMBERS, A. DE MESMAY, T. OPHELDERS*, AND R. ROTMAN, *Constructing monotone homotopies and sweepouts*, accepted for publication in the Journal of Differential Geometry. Cited on page 41, 48, 49, 50, 89

[H] A. DE MESMAY, Y. RIECK, E. SEDGWICK, AND M. TANCER, *Embeddability in* $\mathbb{R}^3$ *is NP-hard*, Journal of the ACM (JACM), 67 (2020), pp. 1–29. An extended abstract appeared in the Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (2018). Cited on page 57, 62, 70, 73

[I] H.-C. CHANG*, J. ERICKSON, D. LETSCHER, A. DE MESMAY, S. SCHLEIMER, E. SEDGWICK, D. THURSTON, AND S. TILLMANN, *Tightening curves on surfaces via local moves*, in Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2018, pp. 121–135. Cited on page 78, 79, 89

[J] E. W. CHAMBERS, A. DE MESMAY, AND T. OPHELDERS*, *On the complexity of optimal homotopies*, in Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2018, pp. 1121–1134. Cited on page 41, 51, 89

[K] V. COHEN-ADDAD, É. C. DE VERDIÈRE, AND A. DE MESMAY, *A near-linear approximation scheme for multicuts of embedded graphs with a fixed number of terminals*, SIAM Journal on Computing, 50 (2021), pp.1–31. An extended abstract appeared in the Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (2018). Cited on page 23, 24, 28, 30, 31

[L] B. A. BURTON, A. DE MESMAY, AND U. WAGNER, *Finding non-orientable surfaces in 3-manifolds*, Discrete & Computational Geometry, 58 (2017), pp. 871–888. An extended abstract appeared in the Proceedings of the 32nd International Symposium on Computational Geometry (2016). Invited to special issue of Discrete & Computational Geometry. Cited on page 57, 61, 62

[M] A. HUBARD, V. KALUŽA*, A. DE MESMAY, AND M. TANCER, *Shortest path embeddings of graphs on surfaces*, Discrete & Computational Geometry, 58 (2017), pp. 921–945. An extended abstract appeared in the Proceedings of the 32nd International Symposium on Computational Geometry (2016). Invited to special issue of Discrete & Computational Geometry. Cited on page 75, 78, 89

[N] V. COHEN-ADDAD* AND A. DE MESMAY, *A fixed parameter tractable approximation scheme for the optimal cut graph of a surface*, in Algorithms-ESA 2015, Springer, 2015, pp. 386–398. Cited on page 23, 24, 26, 29

Other publications

[O] E. W. CHAMBERS, F. LAZARUS, A. DE MESMAY, AND S. PARSA, *Algorithms for contractibility of compressed curves on 3-manifold boundaries*, in Proceedings of the 37th International Symposium on Computational Geometry (SoCG 2021), Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. Cited on page 93

[P] V. COHEN-ADDAD, A. DE MESMAY, E. ROTENBERG*, AND A. ROYTMAN, *The bane of low-dimensionality clustering*, in Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2018, pp. 441–456.

[Q] A. DE MESMAY, *Exposé Bourbaki 1121: Nœuds, mouvements de Reidemeister et algorithmes (d'après Lackenby)*, Astérisque, (2019).

[R] É. COLIN DE VERDIÈRE AND A. DE MESMAY, *Testing graph isotopy on surfaces*, Discrete & Computational Geometry, 51 (2014), pp. 171–206. Preliminary version in Proceedings of the Thirtieth Symposium on Computational Geometry, 2012. Cited on page 2, 8

[S] É. COLIN DE VERDIÈRE, A. HUBARD, AND A. DE MESMAY, *Discrete systolic inequalities and decompositions of triangulated surfaces*. Discrete & Computational Geometry, 53 (2015), pp. 587–620. Preliminary Version in Proceedings of the Twenty-Eighth Symposium on Computational Geometry, 2014. Cited on page 2, 8, 17, 32

[T] B. BURTON, É. COLIN DE VERDIÈRE, AND A. DE MESMAY, *On the complexity of immersed normal surfaces*, Geometry & Topology, Volume 20, 1061–1081. An extended abstract was presented at the European Workshop on Computational geometry in 2014. Cited on page 2, 8

[U] J. R. LEE, A. DE MESMAY, AND M. MOHARRAMI, *Dimension reduction for finite trees in ℓ_1*, Discrete & Computational Geometry, 50 (2013), pp. 977–1032. Preliminary version in Proceedings of the Twenty-Third ACM–SIAM Symposium on Discrete Algorithms, 2012.

PhD Thesis

[V] A. DE MESMAY, *Topics in Low-Dimensional Computational Topology*, PhD thesis, École Normale Supérieure, 2014. Cited on page 1, 2, 7, 8

CHAPTER 1

Introduction

This habilitation thesis presents a compilation of some of the works that I have completed between the defense of my PhD Thesis [V] in 2014 and the present day. My research revolves broadly around the algorithmic and combinatorial questions arising from geometric topology, or in the other direction around how geometric and topological ideas impact the design and analysis of algorithms. As the title of this document highlights, the common thread weaving through most of my works is the systematic exploration of the ***interactions*** between the geometry and topology of a mathematical object on one side, and the complexity of the algorithmic problems surrounding this object on the other side. Typical instances of these interactions arise when asking the following basic questions:

- Pick a combinatorial object that is equipped with a topological structure, for example a graph embedded on a surface. How does this topological constraint impact the combinatorics of the object? How can we leverage the topology to solve algorithmic problems on the graph?
- Pick a topological space and consider the combinatorial and metric structures that it can be endowed with. For example, a surface can be endowed with a continuous metric (e.g., a Riemannian one), or a discrete metric (via an embedded graph). How do these geometries compare to each other? Can we use techniques apparently specific to one world (discrete or continuous) to impact the other world, for example to design algorithms?
- Pick a discrete model of a topological object, for example a simplicial complex triangulating a manifold. What are the best algorithms to solve basic topological questions on this object? For example: are these two topological objects homeomorphic? Is this curve contractible? Are these two knots the same? Can we rely on geometric tools to answer these questions?

These broad questions are at the source of countless fundamental problems, many of which are wide open despite their apparent simplicity. My work revolves mostly around low-dimensional topological objects, such as graphs, surfaces, two-dimensional complexes, knots and 3-manifolds. This is partly motivated by applications: surfaces and their embeddings in $\mathbb{R}^3$ occur naturally in many applications (computer graphics, computer-aided design, geometry processing, materials science, geographic information systems, VLSI design). Furthermore, topology in low dimension possesses a distinct mathematical flavor compared to the higher dimensional case. Indeed, many topological problems tend to become undecidable as soon as the dimension grows past four (for example testing homotopy or homeomorphism). In addition, the topological study of surfaces and 3-manifolds often revolves around geometric ideas, as opposed to the more algebraic theory underlying the topological properties of spaces of higher dimension. This is showcased by the Poincaré conjecture, which was solved using widely different techniques in dimensions five or more (Smale [172]), four (Freedman [70]) and three (Perelman [152]).

In my PhD Thesis [V], I have investigated questions pertaining to the complexity of computing deformations of embedded graphs [R], decompositions of triangulated surfaces [S], and immersed surfaces in 3-manifolds [T]. In my work since then, I have continued exploring the very rich world at the intersection of low-dimensional topology, geometry and algorithms, which has led to 17 publications in peer-reviewed conferences and/or journals in mathematics and theoretical computer science.

We refer to Figure 1.1 for a panorama of some problems covered in this memoir. After the technical preliminaries in Chapter 3, this habilitation thesis is structured so as to showcase three particular lines of work in Chapters 4, 5 and 6, each of them being developed over the course of multiple years and articles. Since these articles have already been published and reviewed, the approach that I have chosen for this document is to avoid being redundant and focus on different aspects of my research than the ones that have already appeared in print. Therefore, the exposition focuses on the ideas and the context instead of the technical aspects, and a particular emphasis is put into telling the story that was weaved between a succession of papers on similar topics. Likewise, I chose whenever possible to present simple alternative constructions instead of the optimal ones.

Cutting surfaces: Shortest Cut Graph and Multicut. In Chapter 4, the main object that we consider is a graph G embedded on surface and we investigate algorithms to undertake one of the most basic operations for surfaces: cutting it into pieces of simpler topology. In the SHORTEST CUT GRAPH problem, the problem is to find a subgraph $H \subseteq G$ of minimal length such that cutting the surface along H yields a disk. This is a basic topological primitive, both in applied settings and for algorithm design, where such a planarization can then be used in order to apply algorithmic techniques specifically designed for planar graphs. The second problem that we look at is a fundamental algorithmic problem: the MULTICUT problem asks, given a graph and a set of pairs of terminals (s_i, t_i) chosen from

of the vertices, to find a minimum subset of edges whose removal disconnects the pairs of terminals. A first observation is that this problem can be recast into a topological one when G is an embedded graph as follows: by basic duality, the MULTICUT problems amounts to computing a shortest subgraph of the dual graph such that cutting along it yields pieces where the pairs of dual terminals are separated. Then both the SHORTEST CUT GRAPH and the MULTICUT problems seek to compute shortest subgraphs with specific topological properties. In Chapter 4, we present approximation schemes for SHORTEST CUT GRAPH and MULTICUT that run in fixed parameter tractable time when parameterized by the approximation factor, the genus of the underlying surface, and the number of terminals in the case of MULTICUT. These algorithms are then complemented by lower bounds: reductions proving that assuming the Exponential Time Hypothesis, these algorithms cannot be improved to yield exact algorithms with similar time dependencies. These lower bounds answer a question of Erickson and Har-Peled [63].

Homotopy height: optimally sweeping a surface. In Chapter 5, we investigate the HOMOTOPY HEIGHT problem, which we can paint in broad strokes as follows: given a surface endowed with some metric (continuous or discrete), what is the best way to *sweep* the surface using a connected curve? Such questions can be thought of in a physical setting: some elastic curve (say, a rubber band) breaks when its length exceeds some fixed value. Is it possible to bring this curve from one specified position to another using a continuous deformation without breaking it? In this problem, the complexity measure that we are trying to minimize is the maximum length that the curve might require during the sweeping. Such optimal sweep-outs have been studied for very different reasons in both the Riemannian and the discrete setting. In the former setting, they provide a way to prove the existence and control the length of closed geodesics, following an argument dating back to Birkhoff. In the discrete world, they have been introduced as a similarity measure between trajectories, and bear interesting similarities with classical notions of graph decompositions such as path-width and branch-width. Our first result is to show that in many different settings, there exists an optimal sweep-out that has nice structural properties: the sweeping curve always stays simple, and always moves in one direction, i.e., it does not backtrack. This is then leveraged to prove that the underlying algorithmic problem lies in the complexity class **NP** and to provide a polynomial-time $O(\log n)$-approximation algorithm. Finally, we observe an intriguing connection with grid minors, which provides us with a fixed-parameter tractable algorithm to compute a specific variant of HOMOTOPY HEIGHT when parameterized by the output.

Hard problems in three dimensions: knots and embeddings. The objective of Chapter 6 is to provide a collection of **NP**-hardness results for topological problems in three dimensions. The common thread in these hardness results is that they all revolve around classical knots and links, i.e., closed curves (or families of those) embedded in $\mathbb{R}^3$. Al-

gorithmic problems occurring in knot theory occupy a peculiar place in the computational landscape, as there are often huge gaps between the best known algorithms and the best computational lower bounds. After providing an introductory outlook on this algorithmic divide, we prove the **NP**-hardness of multiple knot-theoretical problems: detecting whether a given link admits a trivial link as a sublink, computing the unlinking number of a link, and computing a minimal sequence of Reidemeister moves to untangle an unlink, and even an unknot. A common point in all our reductions is that the gadgets revolve around Borromean rings, which seem to provide a useful topological analogue of the structure of a SATISFIABILITY instance. In a second step (which was actually the first step chronologically), we explain how we can also use reduction techniques based on Borromean rings to prove the **NP**-hardness of the seemingly unrelated topological problem $\text{EMBED}_{2\to 3}$, which asks whether a given 2-dimensional complex into $\mathbb{R}^3$. The connection with knot theory occurs through the technical tool of Dehn surgery, which we will introduce before giving the main idea of the proof.

Other works involving surfaces and three-dimensional topology. In Chapters 7 and 8, we present some other results that we have obtained, involving respectively surfaces or three-dimensional objects. The presentation of my work in these chapters in more concise and focuses on explaining the main results and their motivations. In Section 7.1, we present our work on the *universal shortest path metric* question: given a surface S, does there exist a metric on S such that any graph embeddable on S can be embedded with its edges embedded as shortest paths? Such a universal shortest path metric would provide an analogue of the famous Fàry theorem for planar graphs, and provide a geometric approach towards solving a conjecture of Negami on the crossing number of pairs of embedded graphs. In Section 7.2, we provide upper and lower bounds as well as algorithms for the problem of *tightening* a curve or a family of curves on a surface, where we are trying to minimize the number of combinatorial moves used during this tightening process. This tightening process has connections with the operation of simplifying embedded graphs using electrical transformations, and our results also yield results for the latter problem. In Section 8.1, we answer a question of Burton and Makowsky-Mariño, asking whether there exist knots for which any diagram has high treewidth. We show that this is the case for torus knots (among others), by connecting tree decompositions of a knot diagram to topological quantities related to the knot itself. In Section 8.2, we investigate another basic computational problem arising in knot theory: what is the complexity of computing the crossing number of a link? We show that it is **NP**-hard, which is far from telling the end of the story since the best known algorithm is not known to be in **NP**, or anywhere close.

Finally, in Chapter 9, we discuss perspectives that arise as a continuation of the numerous questions investigated in this document. We start by presenting ongoing work undertaken with Master and PhD students in Section 9.1, and then list further ongoing or planned projects in Section 9.2.

This document is mostly self-contained, although familiarity with the main concepts is expected. Apart from the preliminaries in Chapter 3 which regroup notions used in all subsequent chapters, its chapters are independent.

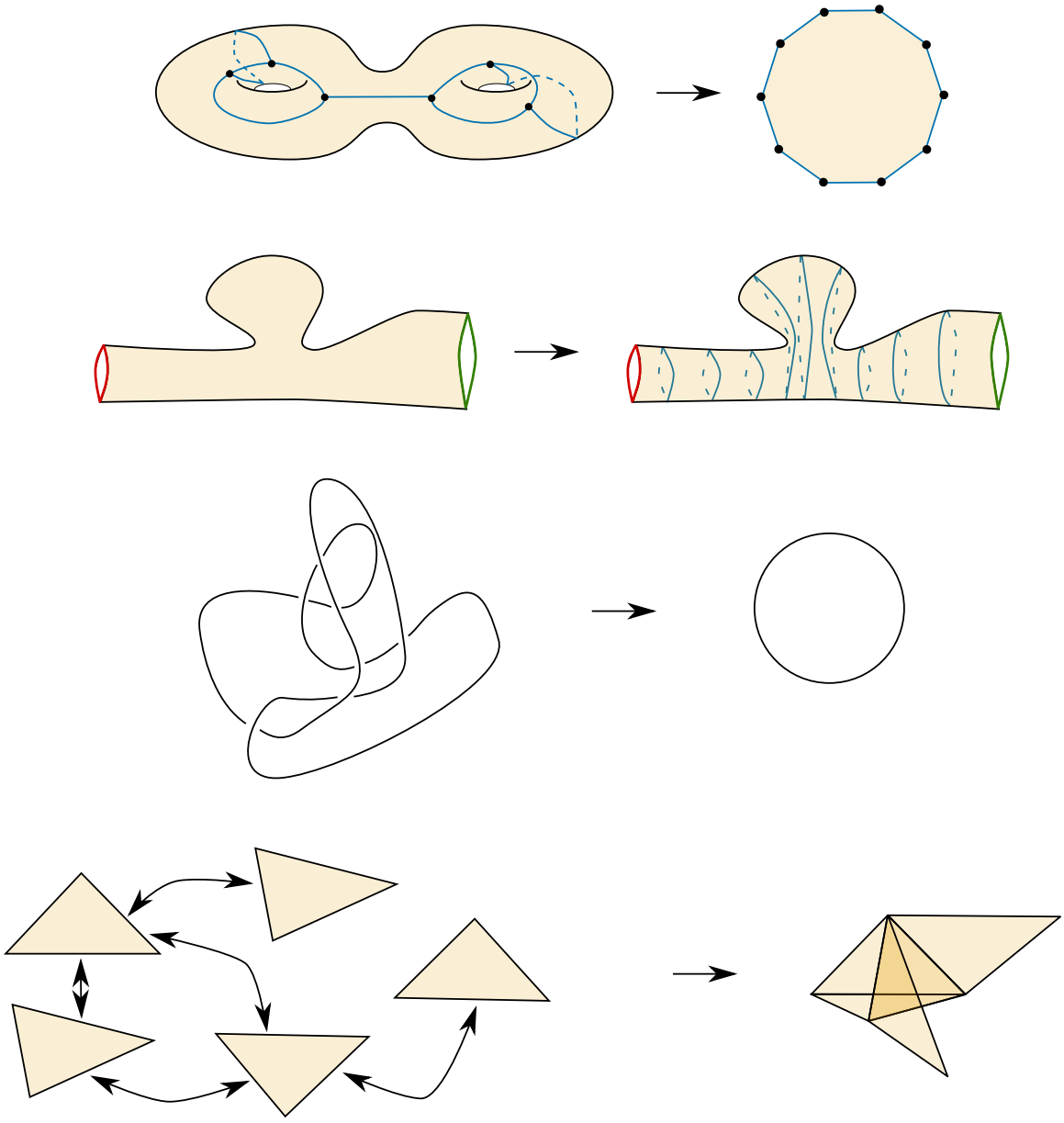

FIGURE 1.1 – Four problems (among others) studied in this habilitation thesis. From top to bottom:

- Given a surface, what is the best way to cut it into a disk?
- Given a annulus endowed with a metric, what is the best way to sweep it, i.e., go from the left boundary to the right boundary by using curves with a minimal maximal length?
- Given a diagram of the unknot, what is the best way to untangle it?
- Given a 2-dimensional simplicial complex, i.e., a family of triangles glued to each other, can we embed this simplicial complex into $\mathbb{R}^3$?

CHAPITRE 2

Introduction en français

Ce mémoire d'habilitation présente une collection de travaux que j'ai complétés depuis la soutenance de ma thèse [V] en 2014. Mes thèmes de recherche traitent des questions algorithmiques et combinatoires qui apparaissent naturellement en topologie géométrique, ainsi que dans l'autre sens des techniques d'origine géométrique ou topologique que l'on peut utiliser dans la conception et l'analyse d'algorithmes. Comme le titre de ce mémoire le met en valeur, il s'agit donc d'explorer de façon systématique les ***interactions*** entre la géométrie et la topologie d'un objet d'un côté, et la complexité des problèmes algorithmiques autour de cet objet de l'autre. Voici quelques exemples de questions issues de ces interactions :

— Prenons un objet combinatoire qui est muni d'une structure topologique, par exemple un graphe planaire ou un graphe plongé sur une surface. Quel est l'impact de cette contrainte topologique sur la combinatoire de l'objet ? Dans le cas d'un graphe plongé, comment peut-on exploiter cette topologie pour résoudre des problèmes algorithmiques sur le graphe ?

— Prenons un espace topologique et étudions les structures discrètes ou métriques dont on peut le munir. Par exemple, une surface peut être munie d'une métrique continue (par exemple riemannienne), ou bien d'une métrique discrète (par le biais d'un graphe plongé). Comment ces deux géométries possibles interagissent-elles ? Peut-on utiliser des techniques venant d'un monde (discret ou continu) pour étudier l'autre ? Plus spécifiquement, les techniques développées dans l'étude de métriques continues peuvent-elles servir à l'algorithmique et la combinatoire des graphes plongés ?

— Prenons un modèle discret d'un objet topologique, par exemple un complexe simplicial qui triangule une variété. Quels sont les meilleurs algorithmes pour identifier les propriétés topologiques de cet objet ? Par exemple : comment reconnaître si deux espaces sont homéomorphes ? Si une courbe est contractile ? Si deux nœuds sont les

mêmes ? Peut-on utiliser des outils géométriques pour répondre à ces questions ?

Malgré leur apparente simplicité, ces questions sont à l'origine de nombreux problèmes fondamentaux, dont beaucoup sont largement inexplorés. Mes travaux tournent principalement autour d'objets topologiques de basse dimension, c'est-à-dire des graphes, des surfaces, des complexes simpliciaux de dimension deux ou trois, des nœuds et des 3-variétés. Cela vient en partie des applications : les surfaces et leurs plongements dans $\mathbb{R}^3$ apparaissent naturellement dans de nombreux domaines (infographie, conception assistée par ordinateurs, traitement géométrique, science des matériaux, systèmes d'information géographique, VLSI). De plus, la topologie de basse dimension a une saveur particulière que ne possèdent pas les espaces de dimension plus grande : en effet, la plupart des problèmes topologiques deviennent indécidables dès que la dimension dépasse quatre (par exemple les problèmes de tester l'homotopie ou l'homéomorphisme). Par ailleurs, l'étude des variétés de basse dimension fait souvent appel à des outils d'origine géométriques, ce qui est différent des techniques plus algébriques qui ont prouvé leur succès en grande dimension. Cela est démontré par exemple par la résolution de la conjecture de Poincaré, qui a nécessité des outils radicalement différents en dimension cinq ou plus (Smale [172]), quatre (Freedman [70]) et trois (Perelman [152]).

Dans ma thèse de doctorat [V], j'ai étudié des questions à propos des déformations de graphes sur les surfaces [R], des décompositions de surfaces triangulées [S] et des surfaces normales immergées dans les 3-variétés [T]. Depuis, j'ai continué à explorer ce monde à l'intersection de la topologie de basse dimension, la géométrie et les algorithmes, ce qui a donné lieu à 17 publications dans des conférences à comité de lecture, et/ou dans des journaux de mathématiques et d'informatique théorique.

La figure 2.1 propose un panorama de quelques problèmes dont ce mémoire traite. Après quelques préliminaires dans le chapitre 3, ce document est structuré autour de trois grands chapitres (4, 5 et 6) qui développent chacun un thème de recherche développé sur de multiples années et donnant lieu à plusieurs publications. Comme les articles en question ont déjà été soumis au processus de revue et de publication, j'ai choisi pour ce mémoire d'essayer d'éviter d'être redondant, et de me concentrer sur des aspects de ces recherches différents de ceux qui ont déjà été publiés. Ainsi, l'on se concentrera davantage sur les idées et le contexte plutôt que sur les aspects techniques, et nous nous attacherons à combler les interstices entre une succession d'articles sur des sujets similaires. Similairement, j'ai préféré dès que possible présenter des constructions simples et pédagogiques, souvent moins efficaces ou complètes que les constructions optimales, plus techniques, qui ont été publiées.

Découper une surface : Graphes de découpe et Multicoupes. Dans le chapitre 4, l'objet d'étude principal est un graphe G plongé sur un surface, et nous nous intéressons à des algorithmes pour réaliser une des opérations les plus basiques pour une surface : découper celle-ci en des morceaux de topologie plus simple. Dans le problème SHORTEST CUT

GRAPH (plus court graphe de découpe), le problème est de trouver un sous-graphe $H \subseteq G$ de longueur minimale tel que découper la surface le long de H donne un disque. C'est une primitive topologique fondamentale, à la fois pour des problèmes d'origine appliquée que pour l'algorithmique : une telle découpe permet ensuite d'exploiter les nombreuses techniques algorithmiques développées spécifiquement pour les graphes planaires. Le second problème que nous regardons dans ce chapitre est un problème classique d'algorithmique : le problème MULTICUT (Multicoupe) demande, étant donné un graphe G et un ensemble de paires de terminaux (s_i, t_i) choisis parmi les sommets, de calculer un ensemble minimal d'arêtes dont la suppression sépare les s_i des t_i. Lorsque le graphe est plongé sur une surface, une observation de base est que ce problème peut être reformulé par dualité comme celui de calculer un plus court sous-graphe $H \subseteq G^*$ tel que découper la surface le long de H donne des morceaux où les paires de terminaux sont séparés. Ainsi, SHORTEST CUT GRAPH et MULTICUT visent tous les deux à calculer des plus court sous-graphes avec des propriétés topologiques particulières. Dans le chapitre 4, nous présentons des schémas d'approximation pour ces deux problèmes qui ont une complexité FPT vis-à-vis du facteur d'approximation, du genre de la surface et du nombre de terminaux pour MULTICUT. Nous présentons ensuite des bornes inférieures qui complémentent ces résultats algorithmiques : essentiellement, nous établissons que l'on ne peut pas avoir d'algorithmes exacts avec des dépendances paramétrées similaires. L'*Exponential Time Hypothesis* nous permet de raffiner ces résultats en donnant des bornées inférieures précises en termes de temps d'exécution. Ces bornes inférieures répondent à une question d'Erickson et Har-Peled [63].

Hauteur d'homotopie : balayer une surface de façon optimale. Le chapitre 5 traite du problème HOMOTOPY HEIGHT (hauteur d'homotopie), que l'on peut décrire de la façon suivante : étant donnée une surface munie d'une métrique soit discrète soit continue, quelle est la meilleure façon de *balayer* cette surface en utilisant une courbe simple ? Plus pratiquement : prenons un élastique qui casse lorsque sa longueur excède une longueur donnée. Est-il possible d'amener cet élastique d'une position de départ à une position d'arrivée en utilisant une déformation continue, sans jamais le casser ? Reformulé sous forme d'un problème d'optimisation, la mesure de complexité que nous minimisons et la longueur maximale que l'élastique pourrait prendre durant le balayage. De tels problèmes de balayages optimaux ont été introduits et étudiés dans des contextes très différents en géométrie riemannienne ainsi qu'en géométrie algorithmique. Dans le premier cadre, ils fournissent une technique pour prouver l'existence de géodésiques fermées et contrôler leurs longueurs. Dans un cadre discret, un balayage optimal peut être utilisé comme une mesure de similarité entre des trajectoires. De plus, il présente de nombreuses similarités avec des décompositions classiques de graphes comme les décompositions en chemins ou en branches. Notre premier résultat est de montrer que dans de nombreux contextes différents, il existe toujours un balayage optimal qui a de très fortes propriétés structurelles : la courbe reste simple à chaque instant, et la balayage est monotone, c'est-à-dire qu'il ne rebrousse jamais chemin.

Ces résultats nous permettent d'établir que le problème algorithmique sous-jacent est dans **NP**, ainsi que d'obtenir un algorithme de $O(\log n)$-approximation qui tourne en temps polynomial. Enfin, nous exploitons la monotonicité pour établir une connexion avec la théorie des mineurs de graphe, qui nous permet d'obtenir un algorithme FPT (en la solution) pour calculer une variante spécifique de la hauteur d'homotopie.

Problèmes difficiles en trois dimensions : nœuds et plongements. L'objectif du chapitre 6 est de décrire une famille de résultats de **NP**-difficulté pour des problèmes topologiques en trois dimensions. Le point commun de ces résultats est qu'ils tournent tous autour de nœuds ou d'entrelacs classiques, c'est-à-dire plongés dans $\mathbb{R}^3$. Les problèmes algorithmiques en théorie des nœuds ont cette particularité qu'il y a généralement un énorme fossé entre les meilleurs algorithmes connus et les bornes inférieures. Après une introduction à ce phénomène, nous établissons la **NP**-difficulté de plusieurs problèmes de théorie des nœuds : détecter si un entrelacs donné admet un sous-entrelacs trivial, calculer le nombre de décroisement d'un entrelacs, et calculer le nombre minimum de mouvement de Reidemeister nécessaire pour démêler le diagramme d'un entrelacs trivial, et même d'un nœud trivial. Un point commun de toutes nos réductions est qu'elles utilisent toutes des gadgets construits à partir des anneaux borroméens, qui semblent fournir un analogue topologique de la structure d'un problème de satisfaisabilité. Dans un second temps (qui était en fait le premier chronologiquement), nous appliquons des techniques de réduction similaires pour prouver la **NP**-difficulté d'un problème d'apparence très différente : celui de tester si un complexe simplicial de dimension deux ou trois se plonge dans $\mathbb{R}^3$. La connexion avec la théorie des nœuds vient d'un outil appelé la chirurgie de Dehn, auquel nous fournissons une introduction avant d'expliquer la preuve.

Autres travaux autour des surfaces et de la topologie en trois dimensions. Dans les chapitres 7 et 8, nous présentons d'autres résultats que nous avons obtenus en lien respectivement avec les surfaces et la topologie en trois dimensions. Dans ces chapitres, la présentation est plus concise et se concentre sur les motivations et les résultats principaux. Dans la section 7.1, nous présentons nos travaux sur la question suivante : étant donnée une surface S, existe-t-il une métrique sur S telle que tout graphe plongeable sur S puisse être plongé tel que ses arêtes soient représentés par des plus courts chemins ? Une telle métrique, dite universelle, fournirait un analogue du célèbre théorème de Fàry pour les graphes planaires, ainsi qu'une approche géométrique pour résoudre une conjecture de Negami sur les nombres de croisements de paires de graphes plongés. Dans la section 7.2, nous fournissons des bornes inférieures et supérieures concernant le problème du *raccourcissement* de courbes ou d'une famille de courbes sur une surface, où l'on essaye de minimiser le nombre d'opérations combinatoires utilisées lors du processus de raccourcissement. Ce processus de raccourcissement a des connexions avec celui de simplifier des graphes plongés en utilisant des *transformations électriques*, et donc nos travaux donnent aussi de

nouveaux résultats pour ce problème. Dans la section 8.1, nous résolvons une question de Burton et de Makowsky-Mariño, qui ont demandé s'il existait des nœuds dont tous les diagrammes avaient une grande largeur d'arborescence. Nous montrons que c'est le cas pour les nœuds toriques (parmi d'autres), en reliant les décompositions arborescents d'un diagramme de nœud avec des propriétés topologiques intrinsèques du nœud lui-même. Dans la section 8.2, nous étudions la complexité du calcul d'un invariant éminemment classique de théorie des nœuds : le nombre de croisements est le nombre minimal de croisements d'un diagramme d'un nœud ou d'un entrelacs donné. Nous montrons que son calcul est **NP**-difficile pour un entrelacs, mais l'écart est encore grand entre cette borne inférieure et le meilleur algorithme connu pour ce problème.

Enfin, dans le chapitre 9, nous évoquons quelques perspectives de travaux dans la continuité de ceux présentés dans ce mémoire. Nous commençons dans la section 9.1 par présenter des travaux en cours avec des étudiants en Master ou en thèse, puis nous introduisons en section 9.2 d'autres perspectives de recherche qui font l'objet de travaux en cours ou futurs.

Ce mémoire introduit l'essentiel des outils sur lesquels il repose, mais il requiert probablement une certaine familiarité avec les concepts utilisés. Les chapitres sont essentiellement indépendants, à part le chapitre 3 qui introduit les notions utilisées dans tout le reste du document.

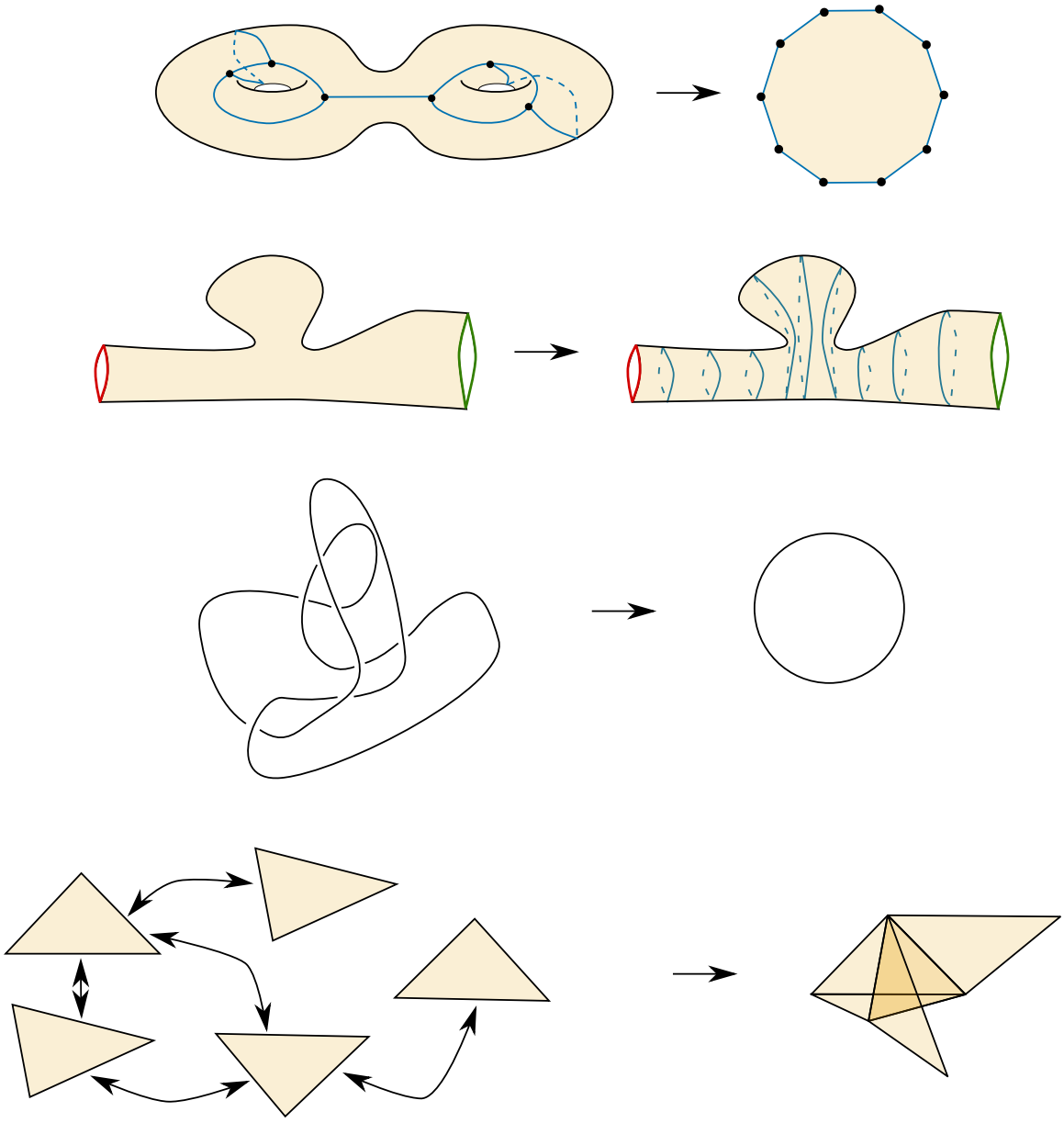

FIGURE 2.1 – Quatre problèmes (parmi d'autres) étudiés dans ce mémoire d'habilitation. De haut en bas :

— Étant donnée une surface, quelle est la meilleure façon de la découper en un disque ?

— Étant donné un anneau muni d'une métrique, quelle est la meilleure façon de le balayer, c'est-à-dire de passer continûment du bord gauche au bord droit en utilisant des courbes de longueur maximale minimale ?

— Étant donné le diagramme d'un nœud trivial, quelle est la meilleure façon de le démêler ?

— Étant donné un complexe simplicial de dimension deux, c'est-à-dire une famille de triangles recollés les uns sur les autres, peut-on plonger ce complexe simplicial dans $\mathbb{R}^3$?

CHAPTER 3

Preliminaries

In this chapter, we introduce the main objects that will be used repeatedly throughout the remaining chapters. The aim of these preliminaries is to be concise and serve as a quick reference while reading the different chapters. For in-depth background on the various objects that will be studied here, we refer the reader to the textbooks of Hatcher [95] and Stillwell [176] (for algebraic topology), Mohar and Thomassen [142] (for graphs on surfaces), Schultens [168] (for 3-manifold theory), Rolfsen [164] (for knot theory), Arora and Barak [7] (for computational complexity) and Cygan et al. [45] (for parameterized algorithms).

Surfaces. A ***surface*** is a topological space where each point has an open neighborhood homeomorphic to $\mathbb{R}^2$. We will also refer to ***surfaces with boundary***, where each point has an open neighborhood homeomorphic to $\mathbb{R}^2$ or to a half-space $\{(x, y) \in \mathbb{R}^2 \mid x \geq 0\}$, the latter points are the ***boundary*** of the surface. By a common abuse of language, we will not always specify "with boundary" when obvious from the context. Compact connected surfaces are classified (up to homeomorphism) by their ***orientability*** and their ***genus***. For surfaces with boundaries, the same classification holds, with the number of boundaries as an additional parameter. Throughout this document, surfaces will always be denoted by S, g will denote their genus, and b will denote the number of boundaries. The genus quantifies intuitively the number of holes in the surface, while the orientability detects whether one can reach the other side of a surface by simply walking on it. The ***Euler characteristic*** of a surface equals $2 - 2g - b$ in the orientable case and $2 - g - b$ in the non-orientable case. Figure 3.1 pictures common examples of low-genus surfaces.

A ***path*** on a surface is a map $p : [0, 1] \to S$. An ***arc*** is a path with endpoints on the boundary of the surface. A ***loop*** based at x is a path such that $p(0) = p(1) = x$. A ***closed curve*** is a path where the two endpoints are the same and thus can be thought of as a map $\mathbb{S}^1 \to S$. A ***curve*** is a path or a closed curve. A path, loop or closed curve is ***simple*** if

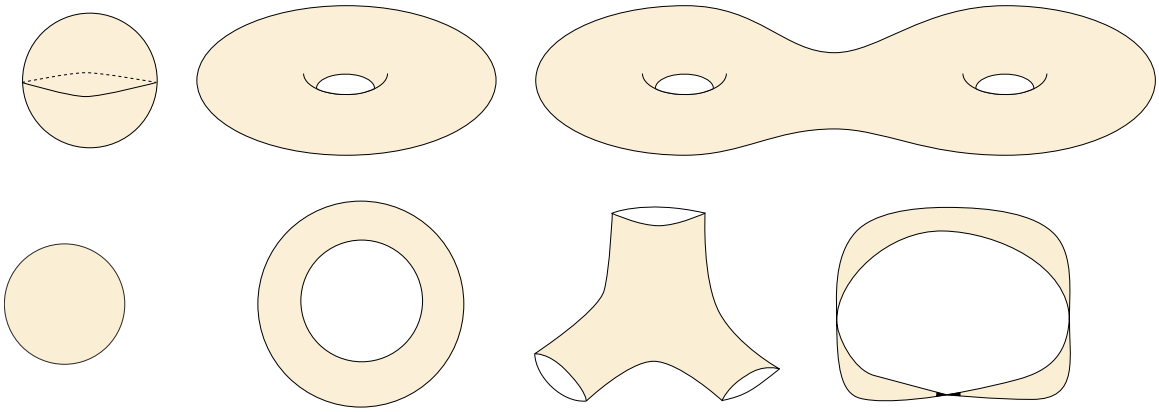

FIGURE 3.1 – Examples of (mostly) orientable surfaces: from top left to bottom right, a sphere, a torus (genus 1), a double torus (genus 2), a disk (genus 0, one boundary), a cylinder (also called annulus, genus 0, two boundaries), a pair of pants (genus 0, 3 boundaries) and a Möbius band (non-orientable, genus 1, one boundary).

the corresponding map is injective. Homotopy captures the intuitive notion of continuous deformation: two closed curves γ_1 and γ_2 are (freely) ***homotopic*** if one can be continuously deformed into the other, i.e., there exists a continuous map $h : [0,1] \times \mathbb{S}^1 \to S$ so that $h(0,\cdot) = \gamma_1$ and $h(1,\cdot) = \gamma_2$. Two simple closed curve are ***isotopic*** if they are homotopic via a homotopy such that $h(t,\cdot)$ is a simple closed curve for any $t \in [0,1]$. Homotopy can similarly be defined for loops with a common basepoint with the constraint that all the intermediate curves also go through the basepoint, and for arcs with various restrictions as to whether and how the endpoints can move on the boundary. The homotopy classes of loops on a surface form a group called the ***fundamental group*** of the surface, denoted by $\pi_1(S)$. We will also encounter the notion of homology in Chapter 6, which is a looser notion of equivalence than homotopy: intuitively, two closed curves are homologous if they cobound a surface. Homology and homotopy generally differ, but on the torus they coincide. The homology classes of closed curves form an abelian group $H_1(S)$, and on the torus $S = \mathbb{S}^1 \times \mathbb{S}^1$, we have $H_1(S) = \pi_1(S) = \mathbb{Z}^2$: closed curves on the torus can be classified by how many times they wind around the torus in the direction of the two $\mathbb{S}^1$s.

Continuous metrics on surfaces. This classification via genus, orientability and number of boundary components only perceives surfaces through a *topological* lens. For many problems studied in this manuscript, we will be also very interested in looking at surfaces through a *geometric* lens, i.e., endowing them with a metric structure. Perhaps the most intuitive metric structure that one can define on a surface is obtained by defining an embedding of the surface into $\mathbb{R}^3$, i.e., an injective map $i : S \hookrightarrow \mathbb{R}^3$. Then the usual metric structure of $\mathbb{R}^3$ naturally defines a length for paths on the surface, and thus a distance function between points. However, it is more standard and fruitful to adopt the following intrinsic

viewpoint: a ***Riemannian metric*** on a surface S is an inner product g_x on the tangent spaces T_xS that varies smoothly when x moves on S. This inner product induces a length on the smooth curves on the surface: if p is a path on S, its length is $\int_0^1 \sqrt{g_x(p'(t), p'(t))}$, and thus we can define a distance by taking the infimum of the length of all paths connecting two points. A ***geodesic*** on a Riemannian surface is a path or a closed curve that is locally minimal, i.e., such that any local perturbation increases its length. Let us emphasize here that a geodesic may not be a shortest path, as the minimality condition is only local. A local invariant of a Riemannian metric is its ***curvature***, which quantifies how locally bent the metric is. One intrinsic way to define it is via the formula $K(p) = \lim_{r\to 0} \frac{12(\pi r^2 - A(p,r))}{\pi r^4}$ where $A(p, r)$ denotes the area of the ball centered at p of radius r. Thus the curvature measures how locally the area of balls differs from the Euclidean area. Surfaces of constant curvature will be of particular interest to us: the geometries in constant positive, zero and negative curvature are respectively called ***spherical***, ***Euclidean*** and ***hyperbolic***. The Gauss-Bonnet theorem is a fundamental formula relating the geometry and the topology of surfaces: it shows that $\int_S K dA = 2\pi\chi(S)$, where K is the curvature and $\chi(S)$ denotes the Euler characteristic. Therefore, the topology restricts strongly the geometry that a surface can be endowed with: for example in the orientable world, only the sphere can be endowed with a spherical metric, only the torus can be endowed with a Euclidean metric, and all the other surfaces can only be endowed with hyperbolic metrics.

Graphs on surfaces and discrete metrics. The previous paragraph described how to endow a surface with a continuous metric. A discrete converse is provided by a graph embedded on a surface. In this manuscript, we will allow for a fairly general class of graphs, allowing loops and multiple edges. Graphs without those will be referred to as ***simple graphs***. A graph $G = (V, E)$ can naturally be seen as a topological space, and an ***embedding*** of G on a surface is an injective map $i : G \to S$, i.e., a drawing of the graph on the surface without crossings. As is customary, we will often identify a graph with its embedding when there is no ambiguity. A ***planar graph*** is a graph embedded in the plane $\mathbb{R}^2$, or equivalently on the sphere $\mathbb{S}^2$. An embedding of a graph defines ***faces*** F which are connected components of the complement $S \setminus i(G)$, and an embedding is ***cellular*** if all the faces are homeomorphic to open disks. In a strong sense, cellular embeddings are the most natural ones, as any embedding on a surface of minimal genus is cellular. A ***triangulation*** is an embedded graph where all the faces are triangles, i.e., are adjacent to three edges. A key invariant of cellularly embedded graphs is the ***Euler characteristic*** $\chi = |V| - |E| + |F|$ which, as the name indicates, equals the Euler characteristic of the surface, i.e. $2-2g-b$, or $2-g-b$ respectively for orientable or non-orientable surfaces. This equality implies that the number of edges of a simple graph satisfies $|E| = |V| + O(g+b)$, and we will refer to $|E|$ as the complexity of the graph, which thus, up to a constant factor, also controls the number of vertices and faces. Cellularly embedded graphs can be described in a purely combinatorial way, independently of the precise location of the vertices and the edges on the surface:

the combinatorial data of the faces, described by the ordered list of edges [1] that they are adjacent to, suffices to reconstruct the homeomorphism class of the embedded graph. Such a data is generally referred to as a ***combinatorial map***. Non-cellularly embedded graphs can be encoded in a similar fashion by also specifying the topology (genus, orientability, number of boundary components of each face, yielding an ***extended combinatorial map***. Some morally equivalent notations are *rotation systems* [142], *fat graph* [82] or *ribbon graphs* [61]. When discussing algorithms and manipulating embedded graphs either as inputs or intermediate objects, they will always be encoded using a combinatorial map, it is easy to see that such a data structure has size $O(|E|)$.

A ***walk*** on an embedded graph is a path on the surface following the edges of the graph. The graph induces a length function on walks by counting the number of edges it uses, which in turn induces a distance function on vertices by considering the shortest walk connecting them. Therefore, embedded graphs provide a discrete way of endowing a surface with a metric, and we call this discrete model a ***combinatorial surface***. When the embedded graph is weighted, the corresponding metric structure can be made more precise by taking into accounts the weights when evaluating the lengths of paths.

We will often employ a different way of using a graph as a metric, relying on duality. A path p on a surface is in ***general position*** with respect to an embedded graph G if the intersection points $p \cap G$ only occur on edges of G, are finite and are transverse, i.e., locally the path always crosses an edge that it intersects. Likewise, two embedded graphs are in general position if they only intersect at edges and these intersections are finite and transverse. An embedded graph induces a length function on paths in general position by simply counting the number of intersections with the graph. Hence it defines a distance function between any two points in the complement of the graph by considering the infimum of the length of all the paths connecting those two points. We call such a metric structure a ***cross-metric*** surface. Cross-metric surfaces were was first introduced by Colin de Verdière and Erickson in [38]. The graph-theoretically minded reader will not be fazed by this definition: it amounts to considering the ***dual graph*** of the embedding, obtained by putting a vertex in each face and connecting (dual) vertices when they are separated by an edge. Then each path in general position can be pushed on this dual graph, and the cross-metric structure boils down to the combinatorial structure of the dual graph. Yet the point of view of cross-metric surfaces has some benefits when we use topological arguments: most importantly the paths that we consider are the paths in general position with respect to a graph, compared to walks on the edges of a graph. The former setting has more "space", and for example in some settings will allow us to consider simple paths, which in the combinatorial setting would use repeated edges and thus would be harder to consider simple. In turn, simple paths are important because they are easily cut along, yielding a new surface with boundary. In this manuscript, we will alternate between the combinato-

1. To be precise, we actually use four ***flags*** per edge, in order to know precisely to which side of the edge and in which direction each face is attached.

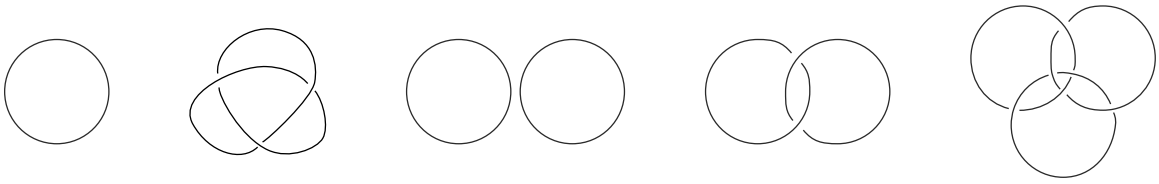

FIGURE 3.2 – Examples of knots and links. From left to right: the unknot, the trefoil knot, the 2-component unlink, the Hopf Link and the Borromean rings.

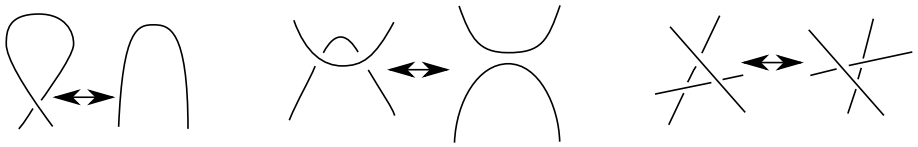

FIGURE 3.3 – The Reidemeister moves I, II and III

rial and the cross-metric point of view freely depending on which is the most natural for the problem at hand. For a rigorous description of combinatorial surfaces and handling of repeated edges, we refer to the habilitation thesis of Francis Lazarus [124].

From many perspectives, graphs on surfaces and Riemannian metrics are two facets of the same coin, which is the study of metric structures on surfaces, and there is quite a wide parallel literature establishing very similar theorems in both settings. The connection can be made formal using for example the two-way street that we designed in [S], which allows for a direct translation of some theorems in one setting to the other one. In recent work, Cossarini [41] provided a correspondence between the discrete and the continuous world with tighter (and optimal) bounds, with the distinction that the continuous metrics that he relies on are Finsler and not Riemannian, that is, the norm that is used on the tangent space is in general not Euclidean.

3-manifolds and knots. A ***3-manifold*** is a topological space where each point has an open neighborhood homeomorphic to $\mathbb{R}^3$. As was the case for surfaces, we will also consider 3-manifolds with boundaries, where points on the boundary have an open neighborhood homeomorphic to a half-space $\{(x, y, z) \in \mathbb{R}^3 \mid x \geq 0\}$. Examples of 3-manifolds include the 3-dimensional sphere $\mathbb{S}^3$, the three-dimensional torus $\mathbb{S}^1 \times \mathbb{S}^1 \times \mathbb{S}^1$ or the ***solid torus***: a torus filled inside, i.e., a space homeomorphic to $D \times \mathbb{S}^1$, where D is a topological disk.

A ***knot*** is an embedding $K \hookrightarrow M$, where M is a 3-manifold. In this thesis, we will almost exclusively consider ***classical knots***, i.e., knots embedded in $\mathbb{R}^3$, or equivalently its compactification $\mathbb{S}^3$. An ***ambient isotopy*** $i : [0, 1] \times \mathbb{S}^3 \to \mathbb{S}^3$ is a continuous family of homeomorphisms such that $i(0, \cdot)$ is the identity (by a slight abuse of language, we often call $i(1, \cdot)$ the isotopy). Two knots K_1 and K_2 are considered equivalent if one can be continuously deformed into the other without creating crossings: formally K_1 and K_2

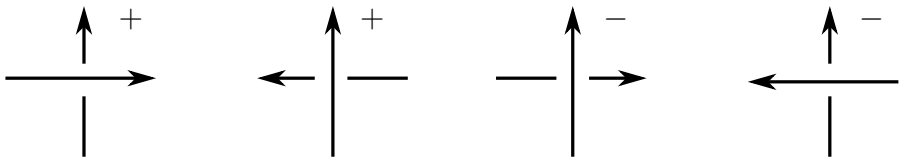

FIGURE 3.4 – The linking number

are equivalent if there exists an ambient isotopy sending K_1 to K_2. The ***trivial knot***, or ***unknot*** is the knot described by the standard embedding of $\mathbb{S}^1$ into $\mathbb{S}^3$, see Figure 3.2, left. This simple definition already hides a great deal of complexity: it is for example not immediate to prove that the trefoil knot is not isotopic to the unknot, or that there exists a non-trivial knot at all. Since the focus of this thesis is on computational aspects of knot theory, we will restrict our attention to ***polygonal knots***, that is, knots of the kind $K : \mathbb{S}^1 \to \mathbb{S}^3$ where the embedding K is a concatenation of segments. This does not lose much generality: every smooth knot is equivalent to a polygonal knot. But it does rule out some pathological behaviors known as ***wild knots***, where some knotting pattern can for example get repeated infinitely many times. The ***connected sum*** of two knots is obtained by removing a small segment on two knots endowed with some orientation and identifying their endpoints so as to preserve the orientation. This operation is well-defined, and every knot can be decomposed in a unique way into a connected sum of prime knots (knots that can not be decomposed further).

A ***link*** is a disjoint union of knots. Links are also considered up to ambient isotopy, and the ***unlink*** on k components is the link obtained by taking k trivial knots contained in k disjoint balls, see Figure 3.2. The ***Hopf link*** is the 2-component link pictured on Figure 3.2, middle right, while the ***Borromean rings*** are the 3-component link pictured on Figure 3.2, right. A ***knot diagram*** or a ***link diagram*** is a generic two-dimensional projection of a knot or link, where one indicates at each crossing point which strand is above and which strand is below. Therefore, a knot diagram is a four-valent planar graph with decorations on the vertices, and the pictures in Figure 3.2 are actually knot and link diagrams. Equivalence of knots or links has a combinatorial interpretation in terms of ***Reidemeister moves***, which are pictured in Figure 3.3: two knot diagrams represent equivalent knots if and only if they can be related by a finite sequence of such Reidemeister moves [157], and a similar theorem holds for links. We will denote by I+ and II+ the Reidemeister moves I and II that increase the number of crossings, and by I- and II- the Reidemeister moves that decrease them. Since they contain knots as a subcase, links are more complicated than knots, but they can be partially distinguished using a very simple invariant called the ***linking number***. For two components of a link, this number can be defined by orienting arbitrarily the two components, and summing the number of positive and negative crossings as defined by Figure 3.4. One can prove that the resulting quantity is indeed a knot invariant by showing that it is invariant under Reidemeister moves. The linking number directly proves that the

Hopf link is not equivalent to the unlink, but is powerless on the Borromean rings, as pairs of components of the Borromean rings have zero linking number (and actually form unlinks when considered in isolation).

Knots and links are important in 3-manifold theory as they naturally provide a host of examples of 3-manifolds: removing a small tubular neighborhood of a classical knot yields a 3-manifold with boundary, and the famous Gordon-Luecke theorem [79] shows that two knots are equivalent if and only if their complements are orientation-preserving homeomorphic. The theorem does not hold for links however (see for example Rolfsen [164, Section 3.A.2]. More subtly, the operations of Dehn filling and Dehn surgery provide a way to define 3-manifolds without boundary from *framed* links. We defer the discussion of these operations to Section 6.2 where they will be used.

Simplicial complexes. A graph cellularly embedded on a surface provides a discrete way of describing a topological space. It is a nontrivial result [105, 154] that any surface can be represented this way. Likewise, one can represent more complicated and higher-dimensional topological spaces using basic building blocks as follows. A k-dimensional simplex is a topological space homeomorphic to the convex hull of $k+1$ points in general position in $\mathbb{R}^{k+1}$. Its ***faces*** are the convex hulls of subsets of these $k+1$ points (note that each face is also a simplex). A ***simplicial complex*** is a collection of simplices K embedded in some ambient space $\mathbb{R}^n$ such that (1) if $\sigma \in K$ and σ' is a face of σ, then $\sigma' \in K$ and (2) if $\sigma, \tau \in K$, then $\sigma \cap \tau$ is a common face to σ and τ. The union of all the simplices in a simplicial complex forms a topological space, which is often identified with the complex itself. The ***vertices*** of K are the 0-dimensional simplices. The isomorphism class of a simplicial complex does not depend on the location of the simplices in $\mathbb{R}^n$, and a simplicial complex can simply be described combinatorially by specifying which subsets of the vertices form simplices. The ***k-skeleton*** of a simplicial complex is the collection of all simplices of dimension at most k. It is an old question whether all topological manifolds can be realized as simplicial complexes, to which the answer is known to be negative, we refer to the book [155] for extensive background on this question and its relatives. Yet, as we already said, all the topological surfaces can be triangulated, and this is also the case for all 3-manifolds, by a result of Moise [143].

Parameterized algorithms and complexity. We consider that the reader is acquainted with basic notions of complexity theory (Turing machines, polynomial-time algorithms, **NP**-hardness) as explained for example in Arora and Barak [7]. In the past decades, two very common approaches have emerged to deal with **NP**-hard problems. The first one is the design of approximation algorithms (see for example Vazirani [180]) and the second one is the design of parameterized algorithms (see for example Cygan et al. [45]). As some of our results in Chapter 4 combine both approaches, we introduce the relevant notions here. The holy grail of approximation algorithms for **NP**-hard problems is a ***Polynomial-Time***

Approximation Scheme (PTAS): an ***approximation scheme*** for a minimization problem is a family of algorithms A_ε parameterized by ε, such that each algorithm A_ε is a $(1+\varepsilon)$-approximation algorithm, i.e., it outputs a solution of cost at most $(1+\varepsilon)OPT$, where OPT is the value of the optimal solution. An algorithmic problem with input of size n and provided with a parameter p is ***fixed parameter tractable*** (FPT) for the parameter p if it can be solved in time $f(p)poly(n)$, where f is a computable function and $poly$ denotes a polynomial function. It has now been clearly demonstrated that FPT algorithms often provide improved runtimes compared to classical ones, even when the size of the parameter is not bounded.

There is a parameterized theory of complexity mirroring the classical theory of **NP**-hardness, for which we will only refer to textbooks (e.g., Downey and Fellows [58]) and just use the fact that under standard conjectures, a problem that is $W[1]$***-hard*** is not FPT, and thus that $W[1]$-hardness (conjecturally) precludes the existence of an $f(p)poly(n)$ algorithm. Typically, $W[1]$-hardness is established by reducing from another $W[1]$-hard problem using a ***parameterized reduction***, that is, a FPT reduction where the size of the reduced parameter is controlled by the input parameter. Typical $W[1]$-hard problems are the problems of determining the existence of a clique of size k in a graph, or an independent set of size k. A lesser known problem is the GRID TILING problem (defined in Section 4.3), which has now been identified as a very convenient problem to establish $W[1]$-hardness of problems in planar graphs (see Cygan et al. [45, Chapter 14]).

In the past decade, a complexity theory that is finer than the usual ones (parameterized or not) has emerged, allowing to prove explicit lower bounds on the complexity of algorithms for certain problems, under stronger conjectures. The ***Exponential Time Hypothesis*** (ETH), which is now established as a key conjecture in the field, stipulates that there is a positive value s such that 3-SAT cannot be solved by an algorithm running in time $2^{sn}(n+m)^{O(1)}$, where n and m denote respectively the number of variables and clauses. As we will see in Chapter 4, the ETH can be used as a starting hypothesis to prove explicit lower bounds for algorithms for numerous problems. We refer to Williams [183] for an introduction to the blooming field of fine-grained complexity.

Graph decompositions. A ***tree decomposition*** of a graph G is a pair (T, χ) where $T = (I, F)$ is a tree, and $\chi = \{\chi_i \mid i \in I\}$ is a family of subsets of V called ***bags*** such that:

1. Every vertex of G appears in some bag of G.
2. For every edge $e = uv$ of G, there exists a bag that contains both u and v.
3. For every vertex v of G, the set of bags that contain v form a connected subtree T_v of T.

The ***width*** of the decomposition is one less than the maximum size of a bag in χ. The ***treewidth*** of a graph G, denoted by $tw(G)$, is the minimum width over all possible tree

decompositions of G. Intuitively, the treewidth of a graph is a quantitative measure of how much a graph looks like a tree: a connected graph has treewidth 1 if and only if it is a tree, while a complete graph K_n has treewidth $n-1$ and a $k \times k$ grid has treewidth k. While originally introduced as a technical tool in structural graph theory, treewidth has emerged as a key parameter in the design of parameterized algorithms, as tree-like structures allow for efficient algorithms based on dynamic programming (see the discussion in Section 8.1 for specific examples in the case of topological problems). We will see in Chapter 4 a converse of this: one can prove precise computational lower bounds for a general family of problems (constraint satisfaction problems) when parameterizing by the treewidth of the underlying graph.

Two other graph decompositions that will be mentioned in this thesis are the ***pathwidth*** and the ***branchwidth***. Pathwidth is defined using path decompositions, which mirror tree decomposition except for the fact that the bags are structured along a path. Therefore pathwidth quantifies how much a graph looks like a path. Branchwidth is a variant of treewidth: the two quantities are always within a constant factor of each other. Branchwidth, and the underlying branch decompositions, display additional structural properties in the case of embedded graphs, and in particular the branchwidth of planar graphs is known to be computable in polynomial time (for treewidth, this is a famous open problem). We refer to Seymour and Thomas [170] for the algorithm and appropriate definitions.

CHAPTER 4

Cutting a surface: approximation schemes and lower bounds

The material in this chapter comes from the three articles [D, K, N] which were co-authored with Vincent Cohen-Addad, Éric Colin de Verdière and Dàniel Marx.

4.1 Introduction

In this chapter, we tell the story of our three articles [D, K, N] which establish approximation algorithms and lower bounds for two cutting problems for graphs embedded on surfaces: the SHORTEST CUT GRAPH problem and the MULTICUT problem (and its close variant MULTIWAY CUT). The main results are summarized in Table 4.1.

The shortest cut graph problem. When one is given a graph embedded on a surface, one of the first things one wants to do with it is to cut the surface open into a disk. One way to do so is by cutting along a subgraph of the original graph: For a graph G with n vertices embedded on a surface S of genus g, a ***cut-graph*** of G is a subgraph $C \subseteq G$ such that cutting G along C gives a topological disk. See an example in Figure 4.1. This is a key primitive for a wide range of applications:

- The first application is perhaps to simply represent the graph, as is done seamlessly throughout this document: once an embedded graph G is cut along a subgraph C to

	SHORTEST CUT GRAPH	MULTI(WAY) CUT
Exact	$n^{O(g)}$ [63]	$n^{O(\sqrt{g^2+gt})}$ [37]
Approx.	$O(\log^2 g)$ in time $O(n \log n)$ [63]	$1+\varepsilon$ in time $f(g,t,\varepsilon)n \log n$ [K]
	$1+\varepsilon$ in time $f(g,\varepsilon)n^3$ [N]	
	$1+\varepsilon$ in time $f(g,\varepsilon)n \log n$ [This chapter]	
Lower b.	NP-hard [63]	NP-hard for $t=3$ [46]
	W[1]-hard w.r.t. g [D]	APX-hard for $t=3$ [46]
	$n^{\Omega(g/log(g))}$ assuming ETH [D]	W[1]-hard w.r.t. g for $t=4$ [D]
		$n^{\Omega(\sqrt{g^2+gt}/\log(g^2+gt))}$ assuming ETH [D]

TABLE 4.1 – Approximate and exact algorithms for SHORTEST CUT GRAPH and MULTICUT problems for graphs of genus g. In the right column, upper bounds are for MULTICUT, lower bounds are for MULTIWAY CUT.

yield a planar graph, one can represent G by drawing the planar graph $G \setminus C$ and the gluing rules on its boundary C to recover the full graph G.

- Cutting a surface into a disk is a cookie-cutter first step in algorithm design, as it allows to solve algorithmic problems on surface-embedded graphs using a simple paradigm: first cut the surface into a disk, then solve the planar problem using the wide array of algorithmic tools available for planar graphs, and finally glue back the disk into a surface and update the solution accordingly.

- In computer graphics or mesh processing, simplifying the topology of a surface is an important step for example in topological noise removal [184], parameterization [54, 83] or texture mapping [128].

In all these applications, it is crucial to control the *length* of the object used to cut the surface into a disk: indeed, if this length stays reasonable, the main features of the embedded graph will be somewhat similar in the remaining planar graph, which ensures that we will not pay too much when re-gluing the surface into its original shape. With this goal in mind, Erickson and Har-Peled [63] introduced the problem of computing the shortest cut-graph:

SHORTEST CUT GRAPH

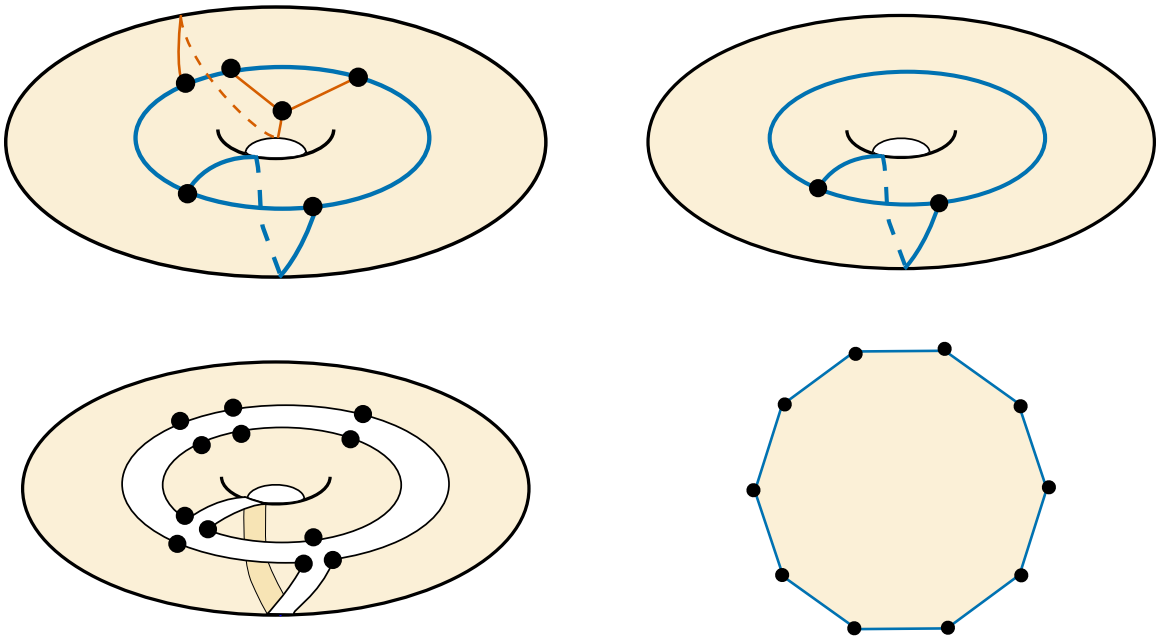

FIGURE 4.1 – A cut graph C of a graph G cellularly embedded on a surface S is a subgraph of G such that cutting S along C yields a disk.

Input: An edge-weighted graph G cellularly embedded on a surface S, and an integer k.
Output: Is there a subgraph $C \subseteq G$ of length at most k such that cutting S along C yields a disk?

They provided a host of results on this problem: they showed that it is **NP**-hard, provided an $n^{O(g)}$ algorithm to compute it exactly, as well as an $O(g^2 n \log n)$ algorithm to compute a $O(\log^2 g)$ approximation. They left as an open problem the fixed-parameter tractability of the problem, that is, whether there exists an algorithm to compute it exactly in time $f(g)n^{O(1)}$. This is a particularly relevant question here as in most practical applications (think about road networks or digitized objects), the genus of the embedded graph is much smaller than the complexity of the graph itself, and thus having an algorithm where the dependency on the genus is isolated could drastically change the efficiency of the algorithm. Furthermore, the shortest cut graph problem contains as a sub-problem the planar Steiner tree problem: intuitively, in order to cut a surface with small, localized handles, into a disk, one must first connect all the handles. The Steiner tree problem famously admits a fixed-parameter algorithm when parameterized by the number of terminals [59]. This gives grounds to be optimistic.

This open problem has kept me occupied for quite some time. Solving it was already an

explicit goal of my PhD thesis, but progress only really started to kick in after my defense. First, with Vincent Cohen-Addad [N], we devised a fixed-parameter tractable algorithm (parameterized by the genus) to compute a $(1+\varepsilon)$-approximation of the shortest cut graph for any ε.

Theorem 4.1.1 ([N])**.** *Let G be an undirected, possibly edge-weighted graph cellularly embedded on a surface S of genus g. For any $\varepsilon > 0$, there exists an algorithm computing a $(1+\varepsilon)$-approximation of the shortest cut graph of G, which runs in time $f(\varepsilon, g)n^3$ for some computable function f.*

While we had good hopes that an exact FPT algorithm was just around the corner, we could not figure it out, and went on a different path to work on a quite different-looking problem.

The multicut problem. The MULTICUT problem is a classic algorithmic problem, generalizing the well-known MIN-CUT problem to an arbitrary number of pairs of terminals: let $G = (V, E)$ be an undirected graph, let T be a subset of vertices of G, called *terminals*, and let R be a set of unordered pairs of vertices in T, called *terminal pairs*. A subset $E' \subseteq E$ is a *multicut* (with respect to (T, R)) if for every terminal pair (t_1, t_2), the vertices t_1 and t_2 lie in different connected components of the graph $(V, E \setminus E')$; see the example in Figure 4.2, left. Then the MULTICUT problem is the following problem:

MULTICUT
Input: An edge-weighted undirected graph G, a subset T of the vertices, a set R of unordered pairs of vertices in T and an integer k.
Output: Is there a multicut with respect to (T, R) of size at most k?

A simpler variant of MULTICUT is MULTIWAY CUT, which is the same problem except that one wants to separate all the terminals instead of a specified set of pairs. MULTIWAY CUT is therefore a special case of MULTICUT when all the possible pairs of terminals are chosen. We will focus on MULTICUT when discussing algorithms, and on MULTIWAY CUT when discussing lower bounds, therefore obtaining the strongest possible results in both cases.

The MULTICUT problem has a very rich history, of which we will only mention the most salient points for this chapter. Already for $|R| = 3$, the problem is **NP**-hard [46], and it is arguably even harder than most hard problems: for many algorithmic problems, one can somewhat get around the hardness by devising efficient approximation algorithms, parameterized algorithms, or by looking at specific instances with more structure, planar or surface-embedded graphs being a prime target here. In contrast, MULTICUT seems to resist all three approaches: it is APX-hard, even for planar graphs [76], so there is little

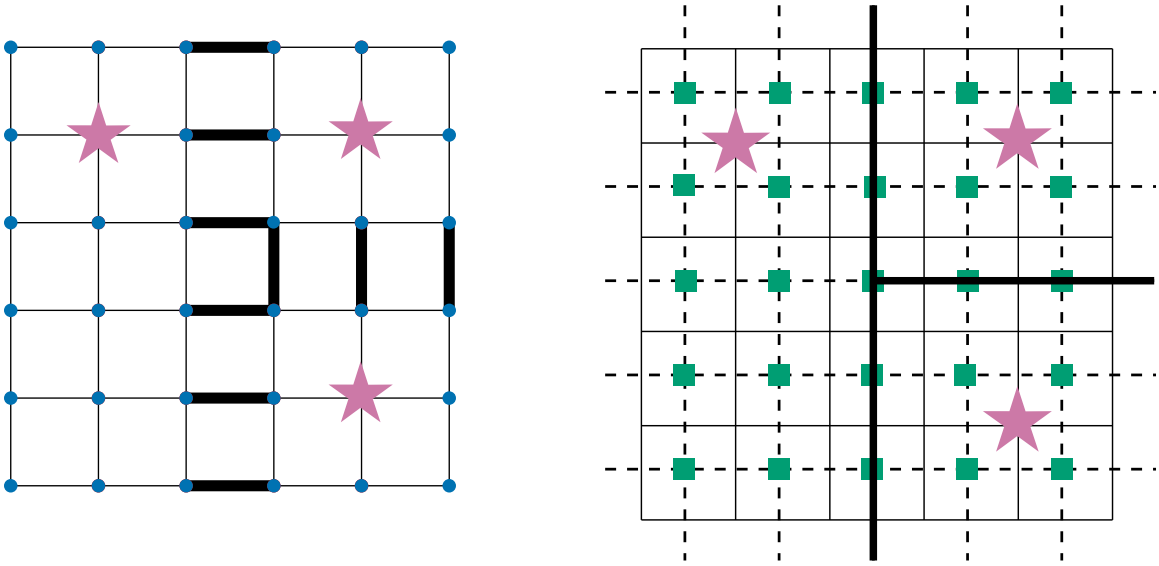

FIGURE 4.2 – A multicut is a set of edges whose removal disconnects a specified set of pairs of terminals (here, all the pairs of stars) (left). For embedded graphs, in the dual, this amounts to finding a shortest graph to cut along so as to separate the pairs of terminals on the surface (right).

hope of devising a polynomial-time approximation scheme. Furthermore, it is $W[1]$-hard when parameterized by the number of terminals, even for planar graphs [135].

Nevertheless, one can still devise algorithms by combining these different approaches to cope with hardness. The first result on this path was obtained by Klein and Marx [110], who showed that MULTIWAY CUT on k terminals could be solved on planar graphs in time $n^{O(\sqrt{k})}$. This was strengthened by Éric Colin de Verdière [37], who tackled the full MULTICUT problem for graphs embedded on a surface of genus g and provided an algorithm running in time $(g+t)^{\alpha(g+t)}n^{\alpha\sqrt{g^2+gt+t+1}}$ for some constant $\alpha > 0$.

In such graph-theoretical cutting problems, the problem is always initially stated as finding a minimum subset of edges whose removal has some desired properties. However, when one is studying embedded graphs, it is always nicer to investigate those under the lens of duality. For example, the dual of a min-cut in a planar graph is a cycle, and the best known algorithms to compute a planar min-cut start by using this dual formulation and then look for a shortest cycle [101]. Likewise, the MULTICUT problem for an embedded graph gets transformed under duality into the problem of finding the shortest transverse graph C^* so that cutting the surface along C^* separates all the required pairs of terminals, see Figure 4.2, right. From that perspective, MULTICUT and the SHORTEST CUT GRAPH problem start looking alike: when considering the terminals as boundaries of the surface, both are problems where one aims to compute the shortest graph cutting the surface into a prescribed *topological* way, in one case the goal is to separate the prescribed boundaries

and in the other case the goal is to obtain a disk. Furthermore, both problems are invariant under homotopy: continuous deformations of the cutting graph do not impact whether it fulfills the objective.

Armed with this analogy, we set on with Vincent Cohen-Addad and Éric Colin de Verdière to develop an analogue of our fixed parameter tractable approximation scheme for the MULTICUT problem, which led to the following result, which stands right on the edge of the precipice formed by all the various hardness results for MULTICUT (see the Introduction of [K] for more details).

Theorem 4.1.2 ([K])**.** *Let G be a undirected, possibly edge-weighted graph embeddable on a surface of genus g, orientable or not. Let n be the number of vertices and edges of G, let T be a set of terminals and R be a set of unordered pairs of T. For any $\varepsilon > 0$, we can compute a $(1+\varepsilon)$-approximation of the minimum multicut of G with respect to (T, R) in time $f(\varepsilon, g, t)n \log n$, where $f(\varepsilon, g, t) = (g+t)^{O(g+t)^3} \cdot (1/\varepsilon)^{O(g+t)}$.*

This turned out to be significantly more technical than we expected. The algorithm is very involved, combining techniques stemming from topology (covering spaces, homotopy, homology), parameterized algorithms (similar to the Dreyfus-Wagner algorithm [59]) and approximation algorithms (we use portals à la Arora [6] and Mitchell [140]) in a delicate way. Exposing the internal workings of this theorem falls squarely outside the scope of this document. However, while designing this algorithm we came up with a much simpler way to approximate the shortest cut graph (Theorem 4.1.1) which also gives improved runtimes. Since it has never appeared in print, it is a good occasion to present it. We will then lay out the multiple additions required in order to obtain Theorem 4.1.2 starting from the same approach.

4.2 Approximation schemes for the Shortest Cut Graph and Multicut problems

Approximating the shortest cut graph. We first explain how to obtain the following strengthening of Theorem 4.1.1.

Theorem 4.2.1. *Let G be an undirected, possibly edge-weighted graph cellularly embedded on a surface S of genus g. For any $\varepsilon > 0$, there exists an algorithm computing a $(1+\varepsilon)$-approximation of the shortest cut graph of G, which runs in time $f(\varepsilon, g)n \log n$ for some computable function f.*

The algorithm follows the outline of many polynomial time approximation schemes (PTAS) for surface-embedded graphs. The main ideas date back to the works of Arora [6] and Mitchell [140] on approximating the Traveling Salesman Problem in Euclidean spaces, and got successively refined into a complete framework for the design of PTAS for planar

graphs (see for example [19, 108, 109]). However, while our initial algorithm [N] followed precisely this framework, involving the use of brick decompositions [108] and some delicate dynamic programming, the version that we expose here is much simpler. Let us denote by OPT the optimal cut-graph. The algorithm is summarized in the following steps.

1. Compute a $O(\log^2 g)$-approximation C of OPT. Cut along it to obtain a disk D.
2. Place a set of points called *portals* at intervals $\eta = |OPT|\varepsilon/g^2$ on $C = \partial D$.
3. Modify OPT into $NearOPT$ by making it go through the portals.
4. Guess the set of portals used by each connected component of $NearOPT$ in D.
5. Solve all the corresponding Steiner tree instances where the terminal set is the set of portals that we guessed.

Let us give details on all these steps and explain how they work together to prove Theorem 4.2.1. In the first step, we use an approximation algorithm of Erickson and Har-Peled [63] to compute an approximate solution C, cutting the surface into a disk D. The length of the boundary of D is, by definition, $O(\log^2 g\, OPT)$, and along the boundary we compute a set of *portals* $\mathcal{P}$, so that all the points on this boundary are at distance at most $\eta := |OPT|\varepsilon/g^2$ of some portal. One can do so using $O(g^2 \log^2 g/(\varepsilon))$ portals. Now, we look at how the solution OPT intersects C. For this, we rely on the following structural lemma:

Lemma 4.2.2 ([63, Lemma 4.1 and 4.2])**.** *The reduced*[1] *shortest cut graph OPT consists of $O(g)$ vertices and edges, and each of its edges can be decomposed in two shortest paths.*

Likewise, the approximate cut graph C has a similar structure: it consists of $O(g)$ vertices and edges, and each edge consists of $O(1)$ shortest paths (we refer to the construction described in [63, Section 6]). Since a pair of shortest paths intersect at most once, C and OPT intersect $O(g^2)$ times. In Step 3, we reroute OPT into another graph $NearOPT$ so that it only crosses C at portals. Since there are $O(g^2)$ crossing points, this increases the length of OPT by at most $O(g^2\eta) = O(\varepsilon|OPT|)$. Therefore, $NearOPT$ is a $(1+\varepsilon)$-approximation of the optimal solution. Denote the connected components of $NearOPT$ in D by $O_1, \ldots, O_k$. Each of them is a tree (otherwise $NearOPT$ would not be a cut graph). Observe that $k = O(g^2)$, and since the number of portals in $\mathcal{P}$ only depends on g and ε, one can guess (i.e., brute-force) the set of portals used by each of the O_i in time $f(\varepsilon, g)$. While any O_i might not be a Steiner tree for its set of portals, replacing it by a Steiner tree makes it shorter. Furthermore, replacing all the O_i by the Steiner trees corresponding to their terminals preserves the fact that they cut the surface into one or more disks: indeed the

1. The reduction means that we consider paths through degree-two vertices as single edges.

worse that can happen is that they run into each other, which makes them cut the surface even more than necessary. Removing the possibly superfluous edges allows us to recover a cut graph. Since $NearOPT$ is a $(1+\varepsilon)$-approximation of OPT and our cut graph is at most the length of $NearOPT$, we obtain a $(1+\varepsilon)$-approximation of the shortest cut graph, as promised. The complexity of our algorithm is dominated by Step 1, which runs in time $O(g^2 n \log n)$, the guesses in step 4 running in time $f(g,\varepsilon)$, and the Steiner tree instances in step 5 can be solved using the Dreyfus-Wagner algorithm [59] which can be made to run in time $2^{O(t)} n \log n$ (see Vygen [182, Introduction]).

Approximating the optimal multicut. Let us now zoom out and highlight what makes this argument work. The key here is that one can compute efficiently a graph with the following two important properties:

1. It is *short*, i.e., it has length $O(|OPT|)$, where the $O(\cdot)$ is allowed to hide dependencies on the parameters.

2. It cuts OPT into a forest.

The first property allows us to place portals and reroute the optimal solution through these portals at a controlled cost. The second property turns the remaining optimization problem into a Steiner tree problem which we can solve efficiently (at least in the parameterized world). In the case of the shortest cut graph, one can conveniently take an approximation of the optimal solution, since, by definition, it satisfies the first property, and the second one follows because any cut graph cuts any other cut graph into a forest. In our analysis, a third property was important, namely that this approximation and the optimal cut graph do not cross too many times, but this turns out not to be that important (as long as the optimal solution has $O(g)$ vertices and edges), as it suffices to reroute each edge a single time if one computes the Steiner trees carefully (see [K, Sections 8 and 9]).

When trying to apply the same framework for the MULTICUT problem, one quickly stumbles into the problem that it is hard to find a graph with these two properties. An approximation C of an optimal multicut OPT need not cut OPT into a forest, see Figure 4.3, top left. Another natural try is to take some shortest tree spanning the terminals, which has the second property, since any remaining cycle would be superfluous. However, there is no guarantee that this tree will be short, see Figure 4.3, top right. Therefore, a major part of the proof of Theorem 4.1.2 is devoted to computing a graph that we call a *skeleton*, which has the two required properties for some $(1+\varepsilon)$-approximate solution.

In order to do that, we fully leverage the leeway allowed by parameterized algorithms. First, we can afford to guess the homeomorphism class of the optimal solution since one can show that its number of vertices only depends on g and t. Even more so, we can afford

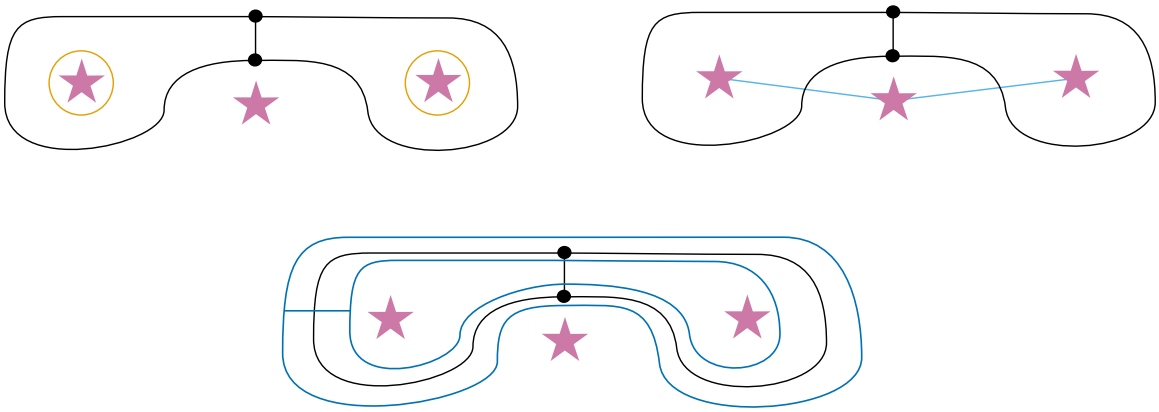

FIGURE 4.3 – OPT is depicted in black in all three figures, an optimal multicut separating the three terminals. An approximate multicut might not cut OPT into a forest (top left). A shortest tree spanning the terminals might be too long (top right). Our algorithm uses a skeleton, that sandwiches OPT between two cycles of roughly the same length and connects the two cycles (bottom).

to treat separately each cycle of the graph formed by this optimal solution so as to cut it. [1] Our approach is to guess the topology and an approximation of the length of each cycle γ in the graph formed by the optimal solution, in order to sandwich it between two cycles γ^L and γ^R of approximately the same length. Then either OPT crosses both γ^L and γ^R, and in this case we can afford to connect γ^L and γ^R using a shortest path since the length of this path will be bounded by OPT, see Figure 4.3, bottom. Or OPT only crosses one of them, say γ^L and then we can push it at a small cost so that it coincides with γ^L. Or it does not cross any of them, but then we can replace it by the shortest cycle in its homotopy class, which we can compute. All in all, we manage to cut every single cycle of OPT, thus cutting it into a forest, while maintaining some control on the length of the skeleton used to do so.

The rest of the algorithm follows the same idea as for the SHORTEST CUT GRAPH problem: one can then put portals at regular intervals on the skeleton, guess the set of portals used by a near-optimal solution, and compute Steiner trees connecting those portals to recover a near-optimal solution. Note however that in contrast to the problem of computing the SHORTEST CUT GRAPH, this last computation might not take place in a disk, and we need to control the homotopy class of the Steiner trees that we compute: whether it leaves a terminal on one side or the other side makes a big difference on the solution. This is achieved by doing the computations in the appropriate *covering space*, which is a tool from algebraic topology allowing us to handle homotopical constraints by considering a larger

1. It actually suffices to cut a subset of all cycles, called an *exhaustive family*, to obtain the second property, leading to a speed-up in the algorithm (but note that cutting all faces does not suffice). We take the simpler approach of cutting all the cycles here, and refer the reader to the paper [K] for more details.

space where we do the computations.

Before concluding this hasty sketch, let us mention two additional technical difficulties: the sandwiching procedure requires carefully optimized algorithms (in particular, to get near-linear time, we rely on the fast computation of planar Multiple Source Shortest Paths [24] in some well-chosen annular covering spaces), and in order to guess the length approximately within the given complexity, we need to have a separate treatment for cycles that are very long.

4.3 Lower bounds

We now come back to the original question of designing an FPT algorithm for the SHORTEST CUT GRAPH problem, armed with our experience with MULTICUT, which, as we now explain, radically changed our expectations.

Exact algorithms to solve both the SHORTEST CUT GRAPH [63] and the MULTICUT [37] problem both start with a similar observation. We want to compute a shortest transverse graph C so that $G \setminus C$ is either a disk or separates the prescribed boundaries. In particular, C, viewed as a (not necessarily cellularly) embedded graph, always has at most t faces, where t is the number of terminals. The number of homeomorphism classes of such graphs (after dissolving degree 2 vertices) only depends on g and t, and since we are designing algorithms parameterized in g and t, we can safely afford to enumerate them all, or, stated differently, to guess the correct one. Therefore, both problems, modulo this observation, can be seen as a subcase of the following more general problem, which I call the SHORTEST MAP EMBEDDING problem:

> SHORTEST MAP EMBEDDING
> **Input:** A graph G cellularly embedded on a surface S, a homeomorphism class M (i.e., an extended combinatorial map) of a graph embedded on S, and an integer k.
> **Output:** Is there a graph H transverse to G, having the topology prescribed by M and at most k crossings with G?

If one allows a FPT preprocessing step, this very general problem contains a wide family of topological problems: computing a shortest non-contractible or non-separating curve (see for example [36] for a survey on algorithms for this problem), a shortest splitting cycle [27], a shortest pants decomposition [S, Section 4], a shortest system of loops [64], as well as the aforementioned shortest cut graph and multicut problems.

Being so general, the shortest map embedding problem must be very hard. Since, as we said earlier, the MULTICUT problem is W[1]-hard on planar graphs when parameterized by the number of terminals [135], it is natural to expect the shortest map embedding problem to be W[1]-hard when parameterized by the complexity of the input map M. Even more,

it incentivizes us to look precisely into the gadgets designed by Marx in [135] to see if they can be used to prove W[1]-hardness for other sub-problems of the SHORTEST MAP EMBEDDING problem, such as the SHORTEST CUT GRAPH problem, or tighter hardness bounds for MULTICUT on higher genus graphs. In a joint work with Vincent Cohen-Addad, Éric Colin de Verdière and Dàniel Marx [D], we succeeded in doing so and obtained the following almost tight bounds.

Theorem 4.3.1 ([D]).

- *The* SHORTEST CUT GRAPH *problem for a graph with n vertices embedded on a surface of genus g is W[1]-hard when parameterized by g. Furthermore, assuming the ETH, there exists a universal constant $\alpha_{CG} > 0$ such that for any fixed integer $g \geq 0$, there is no algorithm solving all the* SHORTEST CUT GRAPH *instances of genus at most g in time $O(n^{\alpha_{CG} \cdot (g+1)/\log(g+2)})$.*
- *The* MULTIWAY CUT *problem for a graph with n vertices and t terminals, embedded on a surface of genus g is W[1]-hard with respect to g if $t \geq 4$. Furthermore, assuming the ETH, there exists a universal constant $\alpha_{MC} > 0$ such that for any fixed choice of integers $g \geq 0$ and $t \geq 4$, there is no algorithm that decides all the* MULTIWAY CUT *instances (G, T) for which G is embeddable on the orientable surface of genus g and $|T| \leq t$ in time $O(n^{\alpha_{MC}\sqrt{gt+g^2+t}/\log(g+t)})$.*

Comparing these lower bounds with the aforementioned best exact algorithms for the SHORTEST CUT GRAPH [63] and the MULTICUT [37] problems, we see that they only differ by a logarithmic factor in the exponent.

At its core, the proof of Theorem 4.3.1 relies on a set of delicately handcrafted gadgets due to Marx [D], which allow us to encode a very general class of constraint satisfaction problems inside the problem of finding a shortest graph with specific topological properties. We first introduce these ***dual cross gadgets***, which depend on a subset $S \subseteq [\Delta]^2$ for some integer Δ.

Dual cross gadgets. A dual cross gadget G^*_S is a planar graph embedded on a disk D, with $4\Delta + 8$ distinguished faces incident to its boundary, which are, in clockwise order, denoted by

$$UL^*, u^*_1, \ldots, u^*_{\Delta+1}, UR^*, r^*_1, \ldots, r^*_{\Delta+1}, DR^*, d^*_{\Delta+1}, \ldots d^*_1, DL^*, \ell^*_{\Delta+1}, \ldots, \ell^*_1.$$

The faces UL^*, UR^*, DR^* and DL^* are considered as (dual) ***terminals*** in such a dual cross gadget.

A dual multiway cut is a set of edges M^* of G^*_S such that cutting the disk D along M^* yields at least four connected components, and the four terminal faces end up in distinct components. We say that a dual multiway cut M^* ***represents*** a pair $i, j \in [\Delta]^2$ if cutting the disk D yields exactly four connected components that partition the distinguished faces into the following classes:

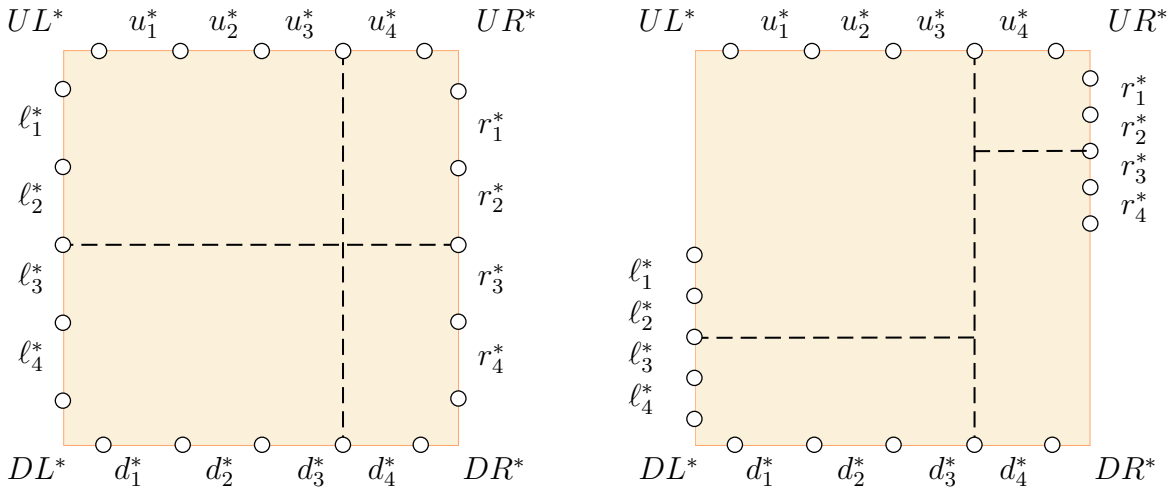

FIGURE 4.4 – Left: a dual multiway cut in a dual cross gadget. Right: A more realistic representation of an actual optimal multiway cut in a dual cross gadget.

$$\{UL^*, u_1^*, \dots, u_j^*, \ell_1^*, \dots, \ell_i^*\} \qquad \{UR^*, u_{j+1}^*, \dots u_{\Delta+1}^*, r_1^*, \dots, r_i^*\}$$
$$\{DL^*, d_1^*, \dots d_j^*, \ell_{i+1}^*, \dots, \ell_{\Delta+1}^*\} \qquad \{DR^*, d_{j+1}^*, \dots, d_{\Delta+1}^*, r_{i+1}^*, \dots r_{\Delta+1}^*\}$$

We refer the reader to Figure 4.4, left, for a (somewhat wrong) representation of a dual cross gadget.

The key properties of the dual cross gadgets are stated in the following lemma.

Lemma 4.3.2 (Dual restatement of [135, Lemma 2])**.** *Given a subset $S \subseteq [\Delta]^2$, we can construct in polynomial time a planar gadget G_S^* with $poly(\Delta)$ unweighted edges and vertices, and an integer D_1 such that the following properties hold:*

i. For every $(i,j) \in S$, the gadget G_S^* has a dual multiway cut of weight D_1 representing (i,j).

ii. Every dual multiway cut of G_S^* has weight at least D_1.

iii. If a dual multiway cut of G_S^* has weight D_1, then it represents some $(i,j) \in S$.

We will not explain the intricate construction of these gadgets here, but it is worthwhile to spend some time to see why obtaining this set of properties is non-trivial. A tempting starting point to design a dual cross gadget is to start with a grid, and play with multiple edges to make some (dual) rows and columns lighter than others, see Figure 4.5, left. This way, we can hope that the rows and columns labeled by S are lighter than others, and thus the dual multiway cuts representing pairs in S are exactly those "crosses" of weight at most D_1. This runs into two issues:

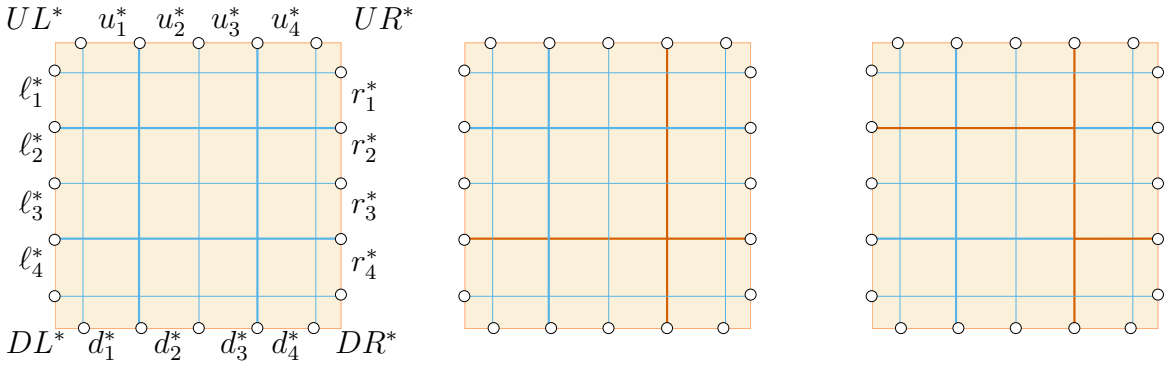

FIGURE 4.5 – A naive attempt at a cross gadget: the rows and columns in bold are lighter than others (left), but it might yield unwanted dual multiway cuts (middle, in orange), or dual multiway cuts that do not represent any pair in S (right, in orange).

1. If S contains pairs (i_1, j_1) and (i_2, j_2) but not (i_1, j_2), the dual multiway cut representing (i_1, j_2) will also be light, see Figure 4.5, middle.

2. If S contains pairs (i_1, j_1) and (i_2, j_2), there is also a light dual multiway cut not representing any pair in S, see Figure 4.5, right.

We invite the reader to ponder about how to tinker the grid so as to avoid these problems. A key idea of Marx [135] is to design the gadgets such that the multiway cuts are not represented by crosses but by more technical shapes to circumvent these two key issues, see Figure 4.4, right.

W[1]-hardness of Shortest Cut Graph. The dual cross gadgets were initially designed to reduce from the following GRID TILING problem, which is known to be W[1]-hard when parameterized by k [45, Section 14.4.1]:

GRID TILING
Input: Integers k, n and k^2 nonempty sets $S_{i,j} \subseteq [n]^2$ for $1 \leq i, j \leq k$.
Output: For each $1 \leq i, j \leq k$, a value $s_{i,j} \in S_{i,j}$ such that:

- If $s_{i,j} = (x, y)$ and $s_{i,j+1} = (x', y')$, then $x = x'$
- If $s_{i,j} = (x, y)$ and $s_{i+1,j} = (x', y')$, then $y = y'$.

Informally, the goal in the grid tiling problem is to find labels on each vertex of a grid that belong to a specified subset that is different for each vertex, and that "agree" horizontally and vertically. See Figure 4.6 for an example of an instance (left), and a valid solution (right).

$S_{1,1}$:	$S_{1,2}$:	$S_{1,3}$:
(1,4)	(5,2)	(1,1)
(3,1)	(1,4)	(2,4)
(2,4)		(3,4)
$S_{2,1}$:	$S_{2,2}$:	$S_{2,3}$:
(2,2)	(3,1)	(2,2)
(1,4)	(1,2)	(2,3)
$S_{3,1}$:	$S_{3,2}$:	$S_{3,3}$:
(1,3)	(1,1)	(2,3)
(2,3)	(1,3)	

$S_{1,1}$:	$S_{1,2}$:	$S_{1,3}$:
(1,4)	(5,2)	(1,1)
(3,1)	**(1,4)**	**(2,4)**
(2,4)		(3,4)
$S_{2,1}$:	$S_{2,2}$:	$S_{2,3}$:
(2,2)	(3,1)	**(2,2)**
(1,4)	**(1,2)**	(2,3)
$S_{3,1}$:	$S_{3,2}$:	$S_{3,3}$:
(1,3)	(1,1)	**(2,3)**
(2,3)	**(1,3)**	

FIGURE 4.6 – A grid tiling instance (left), with a satisfying assignment in boldface (right).

We now explain how the dual cross gadgets can be directly used to reduce GRID TILING to SHORTEST CUT GRAPH, transforming the parameter k into the genus, and thereby showing that SHORTEST CUT GRAPH is W[1]-hard when parameterized by the genus. This reduction was the first idea that ultimately led to the article [D] but was not published in the final version because of its inefficiency (when used with the Exponential Time Hypothesis, see infra). However, its simplicity makes it a good reduction to present here.

We start by defining a cellularly embedded graph G' on a surface S of genus k^2 as follows [1]. We start with a $k \times k$ toroidal grid H, and we embed it on a surface of genus $k^2 + 1$ as follows. Each edge is embedded on a cylinder, and each (degree-4) vertex is embedded on a sphere with four boundary disks. The spheres and cylinders are attached in the natural way, see Figure 4.7. The embedding of the grid H on this surface is not cellular. We fix this as follows: we pick T a spanning tree of H, and for each edge not in T, we subdivide that edge and add a loop going through the middle vertex and around its cylinder (see Figure 4.7, right). This defines a graph G'. It is easy to see that G' is cellularly embedded and furthermore that it has a unique face which is a disk.

Now, starting from a GRID TILING instance with parameter k, we embed it into a SHORTEST CUT GRAPH instance of genus k^2+1 built on G' as follows. The vertices of G' that came from the grid H are replaced by the dual cross gadgets $G^*_{S_{i,j}}$, where i, j designates the vertex coordinates on the grid. The vertices that were added afterwards on the edges not in the spanning tree are replaced by dual cross gadgets G^*_S, where $S = [n] \times [n]$. This concludes the construction and defines a graph G cellularly embedded on S.

We claim that the shortest cut graph of G naturally corresponds to a grid tiling solution, and vice versa. Indeed, since G' is cellularly embedded and G was obtained by replacing

1. We describe the graph G' in very geometric terms, but of course only the combinatorics matter. The reader can directly deduce the combinatorics from the right picture in Figure 4.7

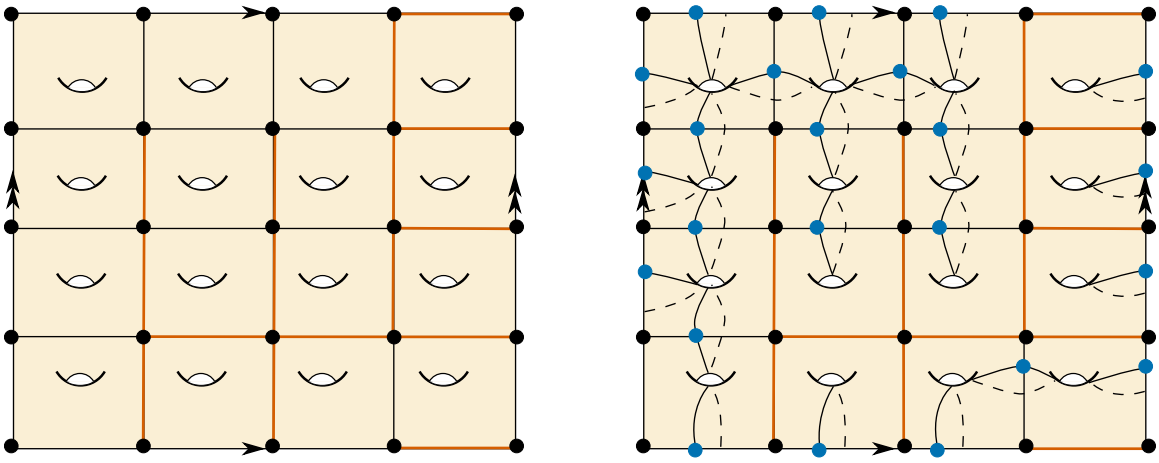

FIGURE 4.7 – Encoding a GRID TILING instance into a SHORTEST CUT GRAPH instance. Left: a non-cellular embedding of a toroidal grid. A spanning tree is colored. Right: Adding loops to edges not in the spanning tree to make the embedding cellular. The final shortest cut graph instance is obtained by replacing each vertex with a dual cross gadget.

its degree 4 vertices by dual cross gadgets, in any cut graph C of G, C must be a dual multiway cut in each of the dual cross gadgets. Therefore, in an optimal cut graph, this dual multiway cut must represent a pair (i, j) in each such gadget (by Lemma 4.3.2), and these labels must agree on adjacent vertices: otherwise one could find a non-contractible cycle by going around the disconnection, contradicting the fact that C is a cut graph. This means that the first labels agree on each column and the second labels agree on each row: we obtain a grid tiling solution. In the other direction, starting from a grid tiling solution, one can choose a dual multiway cut in each of the cross gadgets representing the pairs (i, j) given by this solution. Since the labels agree horizontally and vertically, the endpoints of the dual multiway cut can be connected so as to match for neighboring vertices in the grid. By furthermore adding dual cross gadgets that match on the additional vertices (those coming from the edges not in the spanning tree), we obtain a graph very similar to the graph G', which by construction is a cut graph. The bound on its length follows from the properties of the dual cross gadgets.

Tighter lower bounds. The Exponential Time Hypothesis, a standard computational complexity conjecture that is now ubiquitously used in parameterized complexity, implies that GRID TILING cannot be solved in time $f(k)n^{o(k)}$, for any function computable f, as was proved by Marx (see for example [D]). Therefore, the previous reduction shows that under ETH, SHORTEST CUT GRAPH cannot be solved in time $f(g)n^{o(\sqrt{g})}$. There is still a big gap between this lower bound and the best known exact algorithm running in time $n^{O(g)}$, due to Erickson and Har-Peled [63]. However, the quadratic blow-up in the reduction is inevitable if one starts with GRID TILING since it is already present there. Therefore, in

order to improve the lower bound and obtain the much tighter bounds advertised in Theorem 4.3.1, we departed from GRID TILING and defined the following harder problem:

4-REGULAR GRAPH TILING
Input: Positive integers k, Δ; a four-regular graph Γ on k vertices where the edges are labeled by U, D, L, R in a way that each vertex is incident to exactly one of each label; for each vertex v, a set $S_v \subseteq [\Delta] \times [\Delta]$.
Output: For each vertex v, a value $s_v \in S_v$ such that if $s_v = (i, j)$,

1. the first coordinates of $s_{L(v)}$ and $s_{R(v)}$ are both i, and
2. the second coordinates of $s_{U(v)}$ and $s_{D(v)}$ are both j,

where $U(v)$, $D(v)$, $L(v)$, and $R(v)$ denote the vertex of the graph Γ connected to v via an edge labeled respectively by U, D, L, and R.

The intuition is the following. This problem maintains a structure similar to that of GRID TILING: finding labels on the vertices of a graph, where the labels must belong to a specified set (different for each vertex) and there are local compatibility conditions between adjacent vertices. The problem is richer than GRID TILING since the underlying graph is not restricted to be a grid. Therefore, we can prove the following stronger lower bound:

Theorem 4.3.3 ([D, Theorem 3.1])**.** *1. The* 4-REGULAR GRAPH TILING *problem restricted to instances whose underlying graph is bipartite is W[1]-hard parameterized by the integer k.*

2. Assuming the ETH, there exists a universal constant $\alpha_{\mathbf{GT}}$ such that for any fixed integer $k \geq 2$, there is no algorithm that decides all the 4-REGULAR GRAPH TILING *instances whose underlying graph is bipartite and has at most k vertices, in time $O(\Delta^{\alpha_{\mathbf{GT}} \cdot k/\log k})$.*

The proof of this theorem is based on another key result of Marx [134], where he showed that the fine-grained complexity of *Constraint Satisfaction Problems* (of which our TILING problems are an example) is entirely [1] determined by the treewidth of its graph. Here, a constraint satisfaction problem (CSP) is a generalization of a satisfiability problem defined as follows: it is a triple (V, D, C) where

- V is a set of variables,
- D is a domain of values,

1. Modulo a logarithmic factor in the exponent.

- C is a set of constraints, each of which is a triple of the form $\langle u, v, R\rangle$, where (u, v) is a pair of variables called the scope, and R is a subset of D^2 called the relation.

A solution to a constraint satisfaction problem instance is a function $f : V \to D$ such that for each constraint $\langle u, v, R\rangle$, the pair $(f(u), f(v))$ is a member of R. An algorithm decides a CSP instance I if it outputs true if and only if that instance admits a solution. The primal graph of a CSP instance $I = (V, D, C)$ is a graph with vertex set V such that distinct vertices $u, v \in V$ are adjacent if and only if there is a constraint whose scope contains both u and v.

Marx proved [134] that Constraint Satisfactions Problems whose primal graphs have treewidth tw have computational lower bounds of the kind $n^{\Omega(tw/\log(tw))}$. Grids with n vertices have treewidth $\Theta(\sqrt{n})$, hence the square root in the lower bound of the complexity of GRID TILING. But general 4-valent graphs can have linear treewidth: expander graphs for example have treewidth $\Omega(n)$, and therefore can be used to obtain the strong lower bounds of Theorem 4.3.3.

With Theorem 4.3.3 in hand, the reduction for SHORTEST CUT GRAPH follows a similar blueprint as in the previous paragraph, except that we start with any 4-regular graph instead of a grid. We first embed this graph so that the embedding is cellular and has a unique face (as before, in order to do so we add some loops). Then we replace each degree 4 vertex with a dual cross gadget, modeled along the corresponding set $S_{i,j}$ in the 4-REGULAR GRAPH TILING instance. In the resulting graph G, any shortest cut graph will necessarily realize a dual multiway cut in each dual cross gadget, and thus there is a direct correspondence between a set of labels solving the 4-REGULAR GRAPH TILING instance and the shortest cut graph.

Lower bounds for Multiway Cut. We finish this section with a few comments on how to prove lower bounds for the MULTIWAY CUT problem. The reductions are made technical by the fact that there are two parameters: the genus g and the number of terminals t. We devise two different reductions depending on the regime that we are interested it. On the one hand, when the number of terminals is small, the genus dominates the lower bounds and we aim to prove a lower bound of the type $n^{\Omega(g/\log g)}$. On the other hand, when the number of terminals is at least comparable to the genus, we aim to prove a lower bound of the type $n^{\Omega(\sqrt{gt}/\log(g+t))}$.

The first bound can be obtained rather directly by a reduction from 4-REGULAR GRAPH TILING: here again, we start from the corresponding graph, and replace each vertex by a cross gadget. [1] We identify the four terminals of all the cross gadgets, so that the whole resulting graph has only four terminals. Then, any multiway cut must separate these four ter-

1. Cross gadgets are the duals of dual cross gadgets. As explained in the introduction, in some sense MULTIWAY CUT is dual to the cutting problems à la SHORTEST CUT GRAPH, and thus the gadgets used in our reductions also differ by duality.

minals in all the cross gadgets, and from that we can recover a solution to the 4-REGULAR GRAPH TILING problem.

The second bound requires a bit more work, and to obtain the correct bi-dimensional lower bound we work directly from Constraint Satisfaction Problems and not from 4-REGULAR GRAPH TILING. The main idea is the following: the hardness of 4-REGULAR GRAPH TILING relied on reducing from Constraint Satisfaction Problems built on expander graphs. Here, in order to obtain the correct trade-off between the genus and the number of terminals, we use expander graphs on g vertices, where each vertex has been replaced by a grid of size $\delta \times \delta$, where $\delta = \sqrt{t/g}$. This construction originates from a foundational paper of Gilbert, Hutchinson and Tarjan [78] on separators in embedded graphs. One can show that such graphs have treewidth $\Theta(\sqrt{gt})$, and thus reducing from such instances provides lower bounds of the correct order of magnitude.

CHAPTER 5

Homotopy height

The material in this chapter comes from the three articles [C, G, J] which were co-authored with Therese Biedl, Erin W. Chambers, Gregory Chambers, David Eppstein, Tim Ophelders and Regina Rotman.

5.1 Introduction

Suppose that you are given an elastic bracelet which breaks when it is stretched above some maximal length L, and you wonder whether there is enough wiggling room to put on the bracelet around your hand and reach your wrist, as in Figure 5.1. One way to model this problem is to cut off the hand at the wrist, therefore obtaining a topological disk which is endowed with some metric structure. [2] The bracelet problem then amounts to deciding whether one can find a homotopy from the boundary of this disk to a point, where at no moment the length of the curve is larger than L.

In a disk D with a Riemannian metric, we are therefore looking at the following quantity:

2. We view the bracelet as being very elastic and always sitting on the two-dimensional space defined by the hand. Therefore, the problem only depends on the metric structure of the hand and not on its embedding in $\mathbb{R}^3$

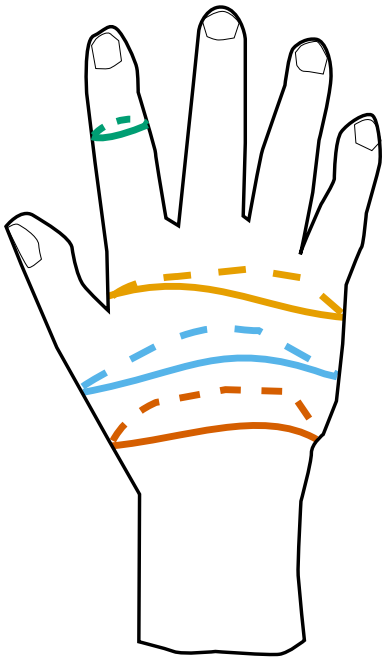

FIGURE 5.1 – Fitting a bracelet around a hand.

$$HH(D) := \inf_{h \in \mathcal{H}} \max_{t \in [0,1]} ||h(t,\cdot)||,$$

where $||\cdot||$ denotes the Riemannian length, and $\mathcal{H}$ denotes the set of homotopies $h : [0,1] \times \mathbb{S}^1 \to D$ where $h(0,\cdot) = \partial D$ and $h(1,\cdot)$ is a trivial curve. If this ***Homotopy Height*** is at most L, we can put on the bracelet, otherwise the bracelet will necessarily break, even if the target position ∂D is much smaller than L.

Similarly, in a disk D which is endowed with a planar triangulation [1] G, we can look at the following quantity:

$$HH(D) := \min_{h \in \mathcal{H}} \max_{t \in [0,n]} ||h(t,\cdot)||,$$

where $||\cdot||$ denotes the length of a walk on the graph G, and $\mathcal{H}$ denotes the set of ***discrete homotopies*** $h : [0,n] \times \mathbb{S}^1 \to D$ where $h(0,\cdot) = \partial D$ and $h(1,\cdot)$ is a trivial curve. Here, a discrete homotopy is a sequence of walks on G, such that two adjacent walks $h(k,\cdot)$ and $h(k+1,\cdot)$ differ by either a ***spike/unspike*** or a ***triangle flip*** (see Figure 5.2). Such a discrete homotopy is a natural discrete analogue of the continuous notion of homotopy used in the Riemannian case above. As before, if the ***Homotopy Height*** is at most L, we can put on the bracelet, otherwise the bracelet will break.

1. We restrict our attention to triangulations for simplicity, but similar definitions exist for any cellularly embedded graph (see Section 5.4).

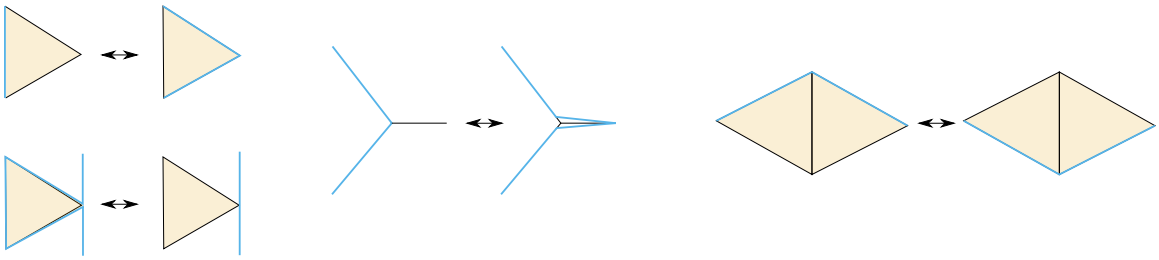

FIGURE 5.2 – The discrete moves in a homotopy: face-flips (left) and spikes/unspikes (middle). Edge-slides (right) will be added in Section 5.4.

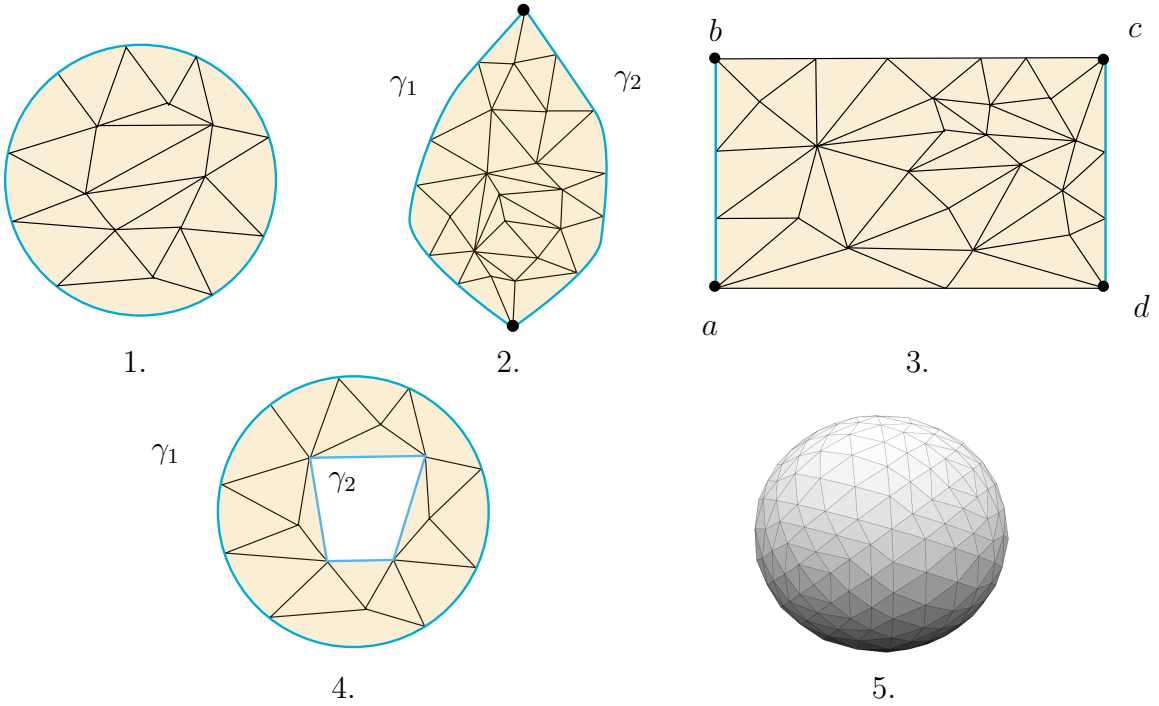

FIGURE 5.3 – Five variants of homotopy height.

One can also define homotopy height in other settings where one wants to quantify how much a curve needs to be stretched to go from a starting position to an ending position using a continuous deformation. Possible settings include (see Figure 5.3):

1. A graph embedded on a disk and we are trying to contract the boundary of the disk to a point (the case discussed above).

2. A graph embedded on a disk, where the boundary is partitioned into two edge-disjoint paths γ_1 and γ_2 and we are trying to move continuously between γ_1 and γ_2 using paths with fixed endpoints. This is the original version of HOMOTOPY HEIGHT as defined in [30].

3. A graph embedded on a disk, with four marked vertices on the boundary a, b, c and d and we are trying to move continuously between ab and cd, using paths with endpoints moving respectively on bc and ad. This variant is often called HOMOTOPIC FRÉCHET DISTANCE [26, 89] since it is closely related to the Fréchet distance [4].

4. A graph embedded on an annulus, and we are trying to move continuously between one boundary and the other one.

5. A graph embedded on a sphere, and we are trying to sweep the sphere by moving continuously from *some* (unspecified) point to some other point. In this variant, we need to add a topological constraint to enforce non-triviality, for example by enforcing that each triangle is flipped exactly once algebraically. [1]

By a slight abuse of language, we will call in this chapter all of these variants HOMOTOPY HEIGHT, since most of our results apply equally well to all of these settings. However, let us insist that none of these settings allows for curves that go outside of the region between the starting and the ending curve (i.e., the starting and ending curves always form the boundary of the underlying manifold). As we shall see later (Figure 5.5), the problem is significantly harder when curves are allowed to escape their starting and ending positions.

The computational problem corresponding to all these variants is the following:

HOMOTOPY HEIGHT
Input: A triangulation G and an integer L.
Output: Is the homotopy height of G (using one of the variants defined above) at most L?

Throughout this chapter, we call a homotopy realizing the minimum an optimal homotopy. [2] The mathematical and computational problems surrounding this homotopy height are of great interest, both in the Riemannian and in the triangulated setting.

Discrete Homotopy Height. From a computational perspective, the study of homotopies on surfaces has a rich history and dates back at least to Dehn [51] who provided an algorithm to test whether two closed curves on a surface are homotopic. However, adding a quantitative twist to this question by forcing curves to have at most some fixed length leads to mostly uncharted territories. The problem of computing the homotopy height was

1. Equivalently, this amounts to requiring that the homotopy, viewed as a map $\mathbb{S}^1 \times \mathbb{S}^1 \to \mathbb{S}^2$ has topological degree 1.

2. In the Riemannian case this is an abuse of language as the infimum may not be a minimum. In that case the term refers to an ε-approximate optimal homotopy, for some small ε (most of our theorems will be stated with *open* bounds on the length, see e.g., Theorem 5.2.2, and thus this perturbation is transparent).

introduced by E. Chambers and Letscher [30], in the setting where one looks at homotopies with fixed endpoints sweeping a boundary from its left side to its right side. Their study was motivated by the problem of measuring quantitatively the similarity between two paths, say, in order to cluster such trajectories in a Geographic Information System. In such a context, one might often encounter two paths with common endpoints which are close in Hausdorff distance, but if there is a mountain inbetween, the homotopy height can be arbitrarily big. Therefore, homotopy height is a more discriminating way to compare paths. Unfortunately, the main result of Chambers and Letscher [30] is erroneous and for a long time essentially nothing was known on this problem. It was open whether in was **NP**-hard, but also simply whether it was in **NP**! Indeed, since discrete homotopies are allowed to backtrack at will, there is no obvious non-deterministic algorithm that can guess an optimal discrete homotopy, as it might consist of exponentially many steps. A few years later, Har-Peled, Nayyeri, Salvatipour and Sidiropoulos [90] provided a polynomial-time algorithm to compute an $O(\log n)$ approximation to the homotopy height in the setting of paths with common endpoints on the boundary of the disk, which is based on a delicate construction of a homotopy using iterated planar separators.

Another motivation for studying homotopy height comes from the field of *graph searching*. This field broadly encompasses a variety of games played on graphs, where a set of k *cops* tries to catch a fleeing *robber*, under various rules. We only sketch some aspects of the theory and refer to the survey of Bienstock [13] for a more extensive overview. A main flavor of these games is that the existence of a winning strategy for the cops or the robber directly translates into structural properties of the underlying graph. For example, in one of the main variants, a winning strategy for a set of k cops implies that the treewidth of the graph is at most $k + 1$, while a winning strategy for the robber implies that it is at least $k + 2$, see Seymour and Thomas [169]. Such games are even more important in the setting of embedded graphs: in a celebrated breakthrough, Seymour and Thomas [170] used such a game-theoretical viewpoint to devise a polynomial-time algorithm to compute the branch-width of a planar graph. Our HOMOTOPY HEIGHT problem bears similarity with graph searching games, since the bracelet sliding can be abstracted as a set of cops trying to catch an invisible robber. Compared to the main variants, the main difference here seems to be a constraint of connectedness: the fact that the bracelet is a simple curve [1] enforces connectivity of the two parts of the surface that it separates. While graph searching games where one side stays connected have been studied [10], the HOMOTOPY HEIGHT problem seems new in this literature. Furthermore, its similarity with branch-width, which enjoys similar connectedness properties (see Seymour and Thomas [170]) and the fact that **NP**-hardness has proved elusive so far make the possibility of a polynomial-time algorithm not as unlikely as one might first be tempted to believe.

To conclude, we also remark that very similar questions have been studied in the context of submodular optimization by Brightweel and Winckler [20], and our work answers one

1. This is not part of the hypotheses, but as we will show later, can be safely assumed.

of their open questions. There are also other notions of height that homotopy height can be connected to, as we will see in Section 5.4.

Riemannian Homotopy Height. Following Gromov [81], there has been in the past decades a strong push to investigate how geometric constraints interact with classical topological primitives, such as homotopy, homology or topological degree, forming a new field called *Quantitative Homotopy Theory*. We refer to the survey of Larry Guth [85] for a highlight of recent questions and solutions. The homotopy height fits nicely into this line of study as it adds a geometric twist to the classical (and trivial!) notion of homotopy on spheres.

The homotopy height has also been studied for a long time (but not under this name) because it provides a tool to control the length of closed geodesics. In order to explain why, let us consider the variant where we sweep a sphere. A ***sweep-out*** of a Riemannian sphere S is a map $f : \mathbb{S}^1 \times \mathbb{S}^1 \to \mathbb{S}$ of topological degree 1 (this amounts to the constraint on flipping triangles mentioned earlier). Intuitively, this just corresponds to a homotopy from a curve to itself, but the topological condition ensures the non-triviality of the homotopy: one has to sweep the whole sphere before coming back. The ***width*** of a sweep-out h is then the maximal length of the curves: $w(h) := \max_{t\in[0,1]} ||h(t,\cdot)||$. The interest in sweep-outs stems from a classical idea of Birkhoff [15] that sweep-outs can be used to find simple closed geodesics on spheres. The process is illustrated in Figure 5.4: starting from a sweep-out S_1, one can apply a ***curve-shortening process*** to the whole family of curves to obtain a new sweep-out S_2. Continuing our analogy with bracelets, one way to think of such a curve-shortening process is to endow each curve with an elastic energy and let it relax for a small amount of time: the elastic energy will move the curve to a position of slightly lower length. If we denote by γ_i the longest curve of the sweep-out S_i, the idea of Birkhoff is that the sequence of curves γ_i converges, as i goes to infinity, to a simple closed geodesic. This not only proves the existence of such a geodesic, but also provides a natural bound on its length: by construction the width of the initial sweep-out provides such an upper bound. Therefore, having good estimates on the minimum width of a sweep-out translates into good estimates on the shortest simple closed geodesic on the sphere, and this led to a long line of investigation on these sweepouts, still active nowadays, see for example the survey of Croke and Katz [43, Section 2.4] and the references therein.

5.2 Structure of optimal homotopies

Our discussion in this chapter focuses on the setting of the disk (discrete or Riemannian), but the techniques apply in the other settings as well. Our initial foray on homotopy height came from trying to figure out how pathological optimal homotopies can be. Such an investigation arises immediately when trying to prove that the HOMOTOPY HEIGHT problem

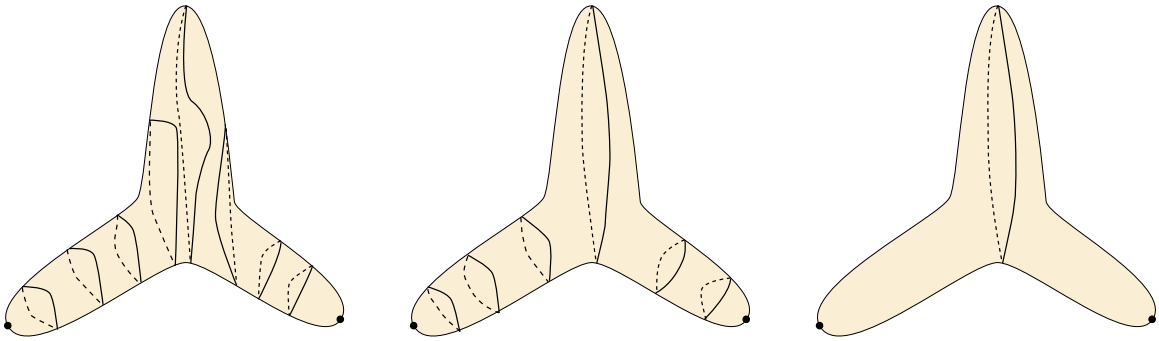

FIGURE 5.4 – Starting from a sweep-out of a Riemannian sphere (left), we can tighten each curve in the family to obtain a new sweep-out (middle). Iterating this process, the longest curve in the family converges to a geodesic (right).

is in **NP**: if we could prove that there always exists an optimal homotopy that sweeps the disk in a polynomial number of steps, this homotopy could be used a **NP** certificate. Two questions arise naturally:

- Does there always exist an optimal homotopy that is an ***isotopy***, that is, where at each step t the curve $h(t, \cdot)$ is simple?
- If the answer to the first question is affirmative, does there always exist an optimal homotopy that is ***monotone***, i.e., that never backtracks?

We define monotonicity as follows: each simple curve γ_t in the disk D bounds exactly one disk D_t. A homotopy is ***monotone*** if for $t_1 < t_2$, the curve $h(t_2, \cdot)$ is contained in the disk D_{t_1}. In a monotone homotopy, curves only move in one direction and never backtrack. In the discrete setting, non-simple curves occur naturally with spikes, therefore we relax the usual definition of simple to what is sometimes called weakly simple: throughout this chapter, a curve is simple if it can be perturbed by an arbitrarily small perturbation so as to be simple [1]. The definition of monotonicity is relaxed likewise so as to allow for arbitrarily small perturbations. With these relaxed notions, it is expected that the answers to these questions is the same in the discrete and Riemannian setting, and as we shall see this is the case.

The first question was answered positively on the Riemannian side by G. Chambers and Liokumovitch [31]. Their proof is based on a beautiful graph-theoretical argument on all the possible resolutions of the crossing points appearing in a homotopy, and applies equally well to the discrete setting. We rephrase their theorem in our setting as follows [2]:

1. Equivalently we could be working in a cross-metric setting where these perturbations happen naturally, i.e. spikes can be realized with simple paths.

2. Their theorem is stated for non-contractible curves, which is definitely not our case here, but this is mainly to avoid issues with orientation, which we did not consider in our definition of homotopy height.

Theorem 5.2.1 ([31, Theorem 1.1])**.** *Let D be a Riemannian or a triangulated disk. Then if there exists a homotopy from the boundary of D to a point with curves of length less than L, there exists an isotopy from the boundary of D to a point, where all the curves have length less than L.*

An affirmative answer to the second question was claimed on the discrete side by E. Chambers and Letscher [30], and on the Riemannian side by G. Chambers and R. Rotman [32] but unfortunately both proofs were later found out to be flawed. Our first contribution was to finally settle this question in a joint effort:

Theorem 5.2.2 ([G, Theorem 1.2])**.** *Let D be a Riemannian or a triangulated disk. Then if there exists a homotopy from the boundary of D to a point, where all the curves have length less than L, there exists a monotone isotopy from the boundary of D to a point, where all the curves have length less than L.*

We also provide a similar theorem for sweep-outs [G, Theorem 1.3]. Likewise, the theorem also holds in the setting of sweeping two paths with fixed endpoints or the annular setting.

We will not provide here a complete proof of Theorem 5.2.2 since such a proof is already included in [G]. However, since [G] is written in a purely Riemannian language, which some discrete-minded readers might not be very accustomed to, we provide a short impressionistic overview of the proof.

Zigzags. The starting point of the proof is to decompose the homotopy we start with in a finite family of monotone homotopies, which we call a ***zigzag***: a zigzag is a family $(h_1, \dots h_n)$ of homotopies so that:

- $h_1, \dots, h_n$ are isotopies.
- $\gamma_0 := h_1(0, \cdot) = \partial D$, $\gamma_{i+1} := h_{i+1}(0, \cdot) = h_i(1, \cdot)$ for $0 < i < n-1$, $h_n(1, \cdot) = p$ where p is a trivial loop at a point p.
- h_i is monotone for odd i, and h_i^{-1} is monotone for even i. Here $\cdot^{-1}$ denotes time-reversal, i.e., for even i, h_i is monotone in the reverse direction.

The height of a zigzag is the maximal length of an intermediate curve. In the discrete setting, the existence of a zigzag of optimal height connecting the starting and ending curve of a homotopy height problem is immediate: indeed, an optimal discrete homotopy can be assumed to be an isotopy by Theorem 5.2.1, and it can be decomposed into a zigzag since each discrete move either goes forward or backward. In the Riemannian case, this is slightly more subtle but can be done at the cost of an arbitrarily small ε additive overhead in the lengths of the curves. Indeed, if a curve wants to move forward at some place, and backward at the some other place, we can do first the forward move for a short period or

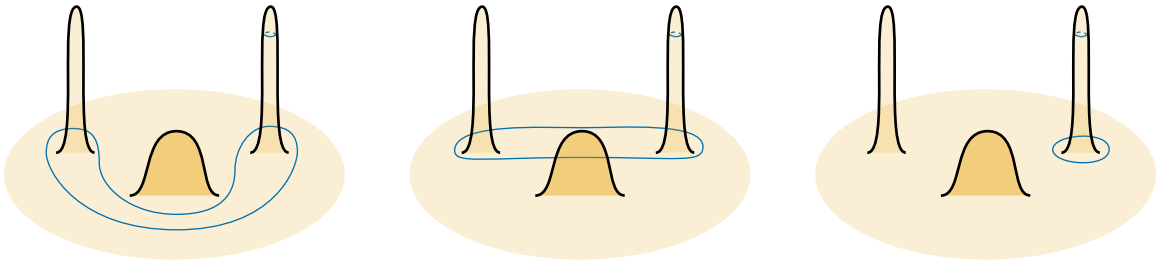

FIGURE 5.5 – An optimal homotopy between the two blue curves picture on the left escapes the annulus that they cobound, and is thus in particular not monotone. Indeed, in order to minimize the maximal length of the curves, it first climbs over the small bump before climbing over the left peak.

time, and only then the backward move. By standard compactness arguments, this will yield a zigzag with a finite number of homotopies.

Once we have decomposed a homotopy into a zigzag, the dream goal is to *shortcut* the zigzag: namely, what the zigzag does in three homotopies h_i, h_{i+1} and h_{i+2}, we would like to do in a single monotone isotopy h'_i that goes from $\gamma_i := h_i(0, \cdot)$ to $\gamma_{i+3} := h_{i+2}(1, \cdot)$ with curves of length less than L. Inductively, we could then transform a zigzag into a single monotone isotopy. This runs into two issues. One is that γ and γ' might cross, making the existence of this monotone isotopy impossible. The second one, more subtle, is that monotone isotopies of optimal length may simply not exist if the curves in the homotopy are allowed to escape the starting curve. This is illustrated in Figure 5.5, which is taken from Chambers and Rotman [32], who attribute it to Liokumovich.

This tricky example shows that a lot of care must be taken in any kind of inductive argument. Our main strategy is to rely on shortcutting arguments such as the following one:

Lemma 5.2.3. *[G, Proposition 2.5] Let A be an annulus with boundaries γ_1 and γ_2 and let δ denote the shortest non-contractible curve within this annulus. Then if there exists a homotopy from γ_1 to γ_2 where intermediate curves have length at most*[1] *L, there exists a homotopy between γ_1 and δ where intermediate curves have length at most*[1] *L.*

The proof consists mostly in observing that a shortest non-contractible curve can be used to shortcut anything that goes through without increasing the length, see Figure 5.6, but one must take care that the resulting curves form a homotopy, which requires some interpolation to avoid discontinuities.

Lemma 5.2.3 applied to the first two homotopies of a zigzag yields the following corollary.

1. In the Riemannian setting, some arbitrarily small offset in the lengths must also be allowed, and therefore this lemma is stated with "less than" instead of "at most" in [G, Proposition 2.5].

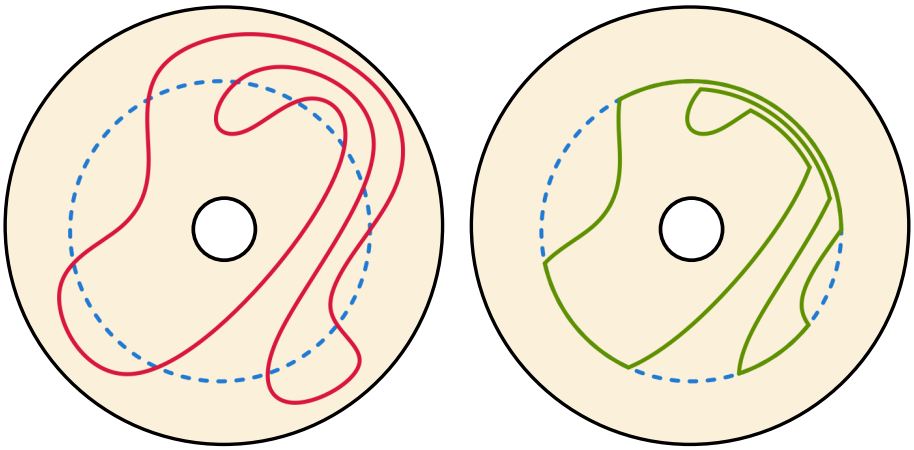

FIGURE 5.6 – In an annulus, shortcutting a curve (red) against a shortest non-contractible curve (dotted blue) yields a new curve (green).

Corollary 5.2.4. *From a zigzag Z where curves have length at most*[1] *L, we can define a new zigzag Z' where curves have length at most*[1] *L and γ_1 is the shortest non-contractible curve in the annulus between γ_1 and γ_2.*

The hope would then be to prove a similar lemma with the other homotopies, i.e., to obtain a zigzag such that γ_i is the shortest non-contractible curve in the annulus between γ_i and γ_{i+1}. This would allow us to shortcut the zigzag at the end: if γ_{n-2} is the shortest non-contractible curve between γ_{n-2} and γ_{n-3}, applying the same lemma in reverse for h_n^{-1} yields a monotone isotopy from p to γ_{n-2}, and we have effectively removed two homotopies from the zigzag.

But already for γ_2, the following issue arises: the shortest non-contractible curve δ_{23} between γ_2 and γ_3 might cross γ_1, destroying any hope of obtaining a monotone isotopy between γ_1 and δ_{23}. In this case, our solution is to use the homotopy between γ_2 and γ_3 to push γ_1 farther, so that it does not cross δ_{23}. This is done via an argument similar to the one used for Lemma 5.2.3, but significantly more involved. We refer to [G] for the details.

5.3 Algorithmic implications

This theorem can be leveraged to obtain complexity results. Indeed, a monotone isotopy only flips at most once each triangle. Furthermore, in an unweighted setting, we can assume that there are never more than L unspikes and L spikes between two triangles flips, since otherwise the curve would be too long at some point. Therefore, there exists an optimal homotopy made of at most $2|T|L$ steps. Such an optimal homotopy can naturally be guessed by a non-deterministic algorithm, since it consists of at most $2|T|L$ curves of length at most L. Therefore we immediately obtain the following corollary.

Corollary 5.3.1. *The* HOMOTOPY HEIGHT *problem on unweighted graphs is in* ***NP****.*

However, when edges are allowed to have weights, the above argument falls short of providing an **NP** certificate. Indeed, it could *a priori* happen that some sequence of edges of exponentially small weight gets spiked an exponential number of times between two subsequent triangles flips, for example by forming a spiral. In a second article with Erin W. Chambers and Tim Ophelders, we showed that this can not happen and were able to prove the following:

Lemma 5.3.2. *[J, Lemma 11] There exists an optimal homotopy such that between two subsequent face flips, each edge is spiked at most three* [1] *times.*

The proof of this lemma hinges on two ideas. The first one is that one can modify a discrete homotopy so that spike moves happen as late as possible, and unspike moves as early as possible, giving what we call a ***reduced*** homotopy where spikes and unspikes do not interfere. The point of reduced homotopies is that they have a nice structure: they start with a sequence of consecutive unspikes, then a sequence of consecutive spikes, then a face-flip, and then the structure repeats. The second idea is that in a sequence of consecutive spikes, we can devise shortcutting arguments not unlike those of Lemma 5.2.3, when an edge gets spiked more than three times. These shortcutting arguments allow us to provide an alternative sequence of consecutive spikes leading to the same curve, where no edge gets spiked more than three times, and thus to prove the lemma. Note that we do not know of any example where an edge needs to get spiked more than once between two subsequent face flips in an optimal homotopy.

From this we directly obtain the complexity result for weighted graphs.

Corollary 5.3.3. *The* HOMOTOPY HEIGHT *problem on edge-weighted graphs is in* ***NP****.*

Our techniques also yield the following polynomial-time approximation algorithm:

Corollary 5.3.4. *We can compute in polynomial time an* $O(\log n)$*-approximation of* HOMOTOPY HEIGHT.

Har-peled, Nayyeri, Salvatipour and Sidiropoulos [89] provided an $O(\log n)$ approximation algorithm for HOMOTOPY HEIGHT in the two variants where one is sweeping a disk with paths starting and finishing on the boundary. In order to obtain Corollary 5.3.4, we leverage our shortcutting techniques to generalize their algorithm to the other settings. For example, in the setting of an annulus with boundaries γ_1 and γ_2, our approximation algorithm proceeds as follows:

1. We compute a shortest path δ between the two boundaries of the annulus.

1. Lemma 11 in [J] states this lemma with an upper bound of four, but the proof actually shows three.

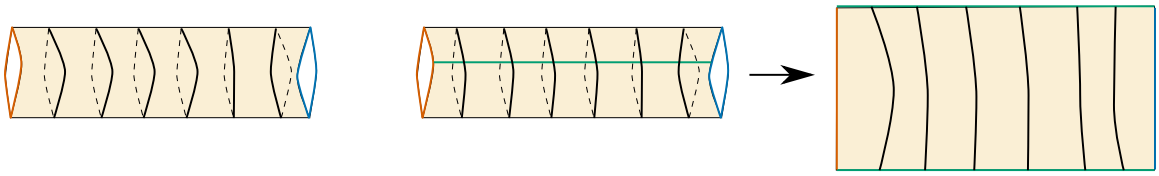

FIGURE 5.7 – To compute an approximation of an optimal homotopy sweeping a cylinder, we first cut along a shortest path and solve the problem in the rectangle. The key property to ensure is that some optimal homotopy has the shape illustrated in the middle picture, i.e., each of the intermediate curve crosses the shortest path exactly once.

2. We cut the annulus along this shortest path, yielding a disk whose boundary is naturally split into four edge-disjoint paths: the starting and ending curve, and two copies of the shortest path δ.

3. We compute an $O(\log n)$ approximation to the corresponding HOMOTOPY HEIGHT problem using the algorithm [89] of Har-peled, Nayyeri, Salvatipour and Sidiropoulos.

4. We glue back the disk into an annulus and obtain from item 3. a homotopy between γ_1 and γ_2 (using portions of δ if we need to connect mismatching endpoints).

The main property that we rely on to prove the approximation ratio is that some approximate solution has the structure output by this algorithm, namely consists of curves which only cross the shortest path δ exactly once, see Figure 5.7. This is proved using similar shortcutting techniques as Lemmas 5.2.3 and 5.3.2.

5.4 Grid-major height

In a subsequent article with Therese Biedl, Erin Chambers, David Eppstein and Tim Ophelders [C], we investigated the connections of homotopy height with other notions of height. The one I would like to highlight here is ***grid-major height***.

Grid Majors. A graph H is a ***minor*** of G if it can be obtained from G by contracting edges, removing edges and removing vertices. Graph minor theory investigates graphs based on the minors that they admit, or do not admit. For example, the Wagner theorem states that a graph is planar if and only if it does not admit K_5 as a minor. A cornerstone of graph minor theory is the grid minor theorem [159], stipulating that there exists a function f so that any graph G of treewidth at least $f(k)$ contains a grid of size $k \times k$ as a minor. One can turn minors into a height parameter by looking at them in the other direction, i.e., investigating graphs based on which grids they are, or are not, a minor of. Since we have turned the usual notion upside down, we introduce the word ***major*** to denote the relation

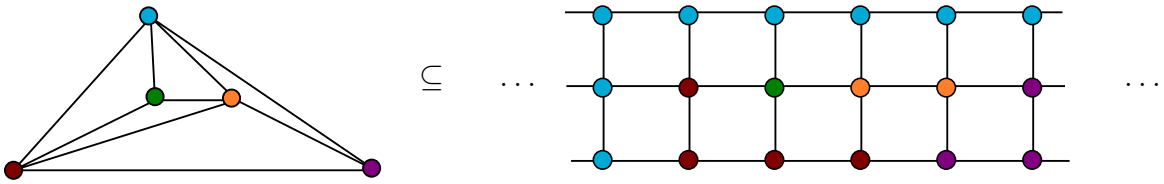

FIGURE 5.8 – The triangulated planar graph is a minor of the grid on the right, as is witnessed by the coloring of the vertices.

opposite to minors, and define the ***grid-major height*** of a graph G as the smallest h so that G is a minor of a $W \times h$ grid for some integer $W > 0$. One way to picture this major relationship is with the following *grid-representation*: a graph G has grid-major height h if one can label all the vertices of the $h \times W$ grid with vertices of G in such a way that (1) every vertex of G appears at least once as a label, (2) for any vertex v, the grid-points labeled by v induce a connected subgraph of the grid, (3) and for any edge uv in G, there is a grid-edge where the ends are labeled by u and v. See Figure 5.8 for an example.

In [C], we proved that this notion is equivalent to a variant of homotopy height with the following rules.

- the homotopy is taken to be between two (unspecified, possibly trivial) paths on the boundary of a face, which we consider as the outer face,
- one additional move, an *edge-slide*, is allowed [1] (see Figure 5.2, right),
- there is a topological condition to avoid triviality, namely that every triangle should be flipped (either via a triangle flip or an edge-slide) algebraically exactly once.

Theorem 5.4.1. *[C, Lemmas 3 and 5] The grid-major height and the homotopy height (with the rules above) of a triangulated planar graph are equal.*

To me, the equality between grid-major height and homotopy height is both natural (in hindsight) and surprising. Let us start with the surprise, which comes from the fact that these are very different parameters from at least two perspectives.

- Homotopy height is a way to quantify the height of an *embedded* graph, since the local rules for homotopies heavily depend on the embedding. On the other side, grid-major height is purely graph-theoretical and completely oblivious to any embedding (note however that it is only finite for planar graphs, since a non-planar graph cannot be a minor of a grid).

1. Since each edge-slide can be decomposed into two face-flips, this new move changes the value of the homotopy height by an additive value of at most one. Furthermore, optimal homotopies with edge-slides can also be assumed to be monotone, see [C, Lemma 4]

- Since minors of minors are minors, grid-major height is trivially closed under minors. In contrast, we would not expect homotopy height to be closed under minors: intuitively, removing an edge in a graph can make shortest paths *longer*, and thus we would expect homotopy height to sometimes increase when taking minors.

On the other hand, the result is quite obvious because the proof proceeds along a very natural path:

- If a triangulated planar graph G has grid-major height at most h, then by definition it can be found as a minor of the $h \times W$ grid, for some $W > 0$. Then in a grid representation, each column of the grid denotes a path in the graph of length at most h, and one can view this collection of paths as a homotopy to sweep the graph. A careful analysis shows that this works, except that this sometimes requires the new edge-slide move.

- In the other direction, a homotopy sweeping a graph G is a collection of paths, where each pair of neighboring paths differ by a fixed set of rules. A natural attempt is to write this collection of paths all next to each other and hope that it yields a grid representation. For this to succeed, we need some careful padding so that all the paths have the same length h, and so that the edge spikes and face flips correspond nicely to edges in the grid representation. But the key difficulty is to prove (2), that is, connectedness of the subgraphs with a common label in the grid. This turns out to be equivalent to monotonicity of the original homotopy, which we can assume by Theorem 5.2.2 and [C, Lemma 4].

This sketch of proof gives us hints as to how to accommodate our surprise. Grid-major height gives us a way to define homotopy height for graphs that are not triangulated, which we can interpret geometrically as follows. As hinted above, the naive definition where triangle flips are replaced with more general face flips is *not* the correct one, since this would not yield a minor-closed parameter. The solution is to look at other curves than the curves on the primal graph of the triangulation. The cross-metric setting is of no help here either: dual curves suffer from the same issue as primal curves, since contracting an edge in the primal graph amounts to removing an edge in the dual graph. But one can instead work with ***nooses*** [1] which are curves on a surface that only intersect the graph at its vertices, and whose length is the number of intersections with the vertices. Nooses in triangulated graphs can be projected to primal curves while preserving their lengths, but in planar graphs with higher-degree faces, they differ because they can cut right through high-degree faces. Furthermore, the length of nooses can only diminish when taking minors. Thus defining an appropriate notion of local moves on nooses (which naturally includes edge-slides) yields a good generalization of homotopy-height to planar graphs that are not triangulated.

1. Nooses originate from structural graph theory, where they form the backbone for the geometric interpretation of branch-width in a planar setting, see [170].

A peculiar corollary of Theorem 5.4.1 is the following.

Corollary 5.4.2. *The problem* HOMOTOPY HEIGHT *can be solved in time fixed-parameter tractable with respect to the output.*

Indeed, grid-major height is trivially closed under taking minors. Thus, via Robertson-Seymour theory [163], the property of having grid-major height at most k is characterized by a finite family of forbidden minors. One can therefore test whether a graph has grid-major height at most k by testing whether it contains any of these forbidden minors, which can be done in polynomial time [162]. Theorem 5.4.1 tells us that this algorithm equivalently computes homotopy height. Note however that we do not know what the forbidden minors are, and thus Corollary 5.4.2 is merely an existence theorem: the actual algorithm is unknown.

Graph Drawing Height. It is a well-known theorem of Fàry [66] that any planar graph can be embedded in the plane so that the edges are embedded as straight lines. Furthermore, this can be achieved so that the vertices of the planar graph are put on lattice points $\mathbb{Z}^2$, and a lot of effort has been devoted to finding the best upper bound on the size of the needed grid (see for example the survey [181]). The graph drawing height investigates this question, but only tries to control one dimension: $Slh(G)$ is defined as the smallest H so that G can be embedded in the plane with straight lines, with vertices on the grid $\{1, \dots, W\} \times \{1, \dots H\}$ for *some* $W > 0$. From many perspectives, this height is a mysterious parameter, in particular it is neither known whether it is computable in polynomial-time nor whether it is **NP**-hard. Furthermore, there are only few known techniques to bound this height from below, most notably relating it to the pathwidth [60, 67] or the outerplanarity [47, 56]. In [C], we also showed that homotopy height can be also used as a tool to bound from below the graph drawing height, yielding sometimes better bounds than those provided by pathwidth and outerplanarity, but we also provided examples where the two quantities differ exponentially.

CHAPTER 6

Hard topological problems in three dimensions

The material in this chapter comes from the three articles [E, H, L] which were co-authored with Benjamin A. Burton, Yo'av Rieck, Eric Sedgwick, Martin Tancer and Uli Wagner.

6.1 Introduction

6.1.1 Hard problems in knot theory

The computational complexity of topological problems depends on the dimension of the ambient manifold in a very sharp manner. Most problems in two dimensions are tractable in polynomial time, and sometimes even linear time, as is for example the case for the problem of testing whether two surfaces are homeomorphic (via the Euler characteristic and an orientability test) or whether closed curves are homotopic [126]. On the other hand, starting in dimension 4, many topological problems, with the notable exception of homology computations, which remain computable efficiently in any dimension, become undecidable. This is for example the case for the problem of deciding whether a closed curve in a manifold is contractible [18, 147], whether the fundamental group of a manifold is trivial [1], or whether two manifolds are homeomorphic [132]. The case of dimension 3 provides an interesting middle ground between these two behaviors: most topological problems in three dimensions are not undecidable, but depending on the problems, the

complexity of the best known algorithms generally ranges from exponential to outright galactic. We first illustrate this phenomenon by surveying the current state of the art of algorithms for three classic computational problems in knot theory.

UNKNOT RECOGNITION
Input: A knot K described by a knot diagram of complexity n.
Output: Is the knot K the unknot?

UNKNOT RECOGNITION is perhaps the most famous problem in computational low-dimensional topology. Proving that it is decidable is already delicate, as first observed by Turing [178]. The first algorithm to solve it was designed by Haken [88], in a seminal article where he invented *normal surface theory*, which is one of the main algorithmic tools in low-dimensional topology. From the point of view of complexity classes, the UNKNOT RECOGNITION problem is known to lie in **NP** [91, 119], as well as in co-**NP** [114, 120], making it most likely to **not** be **NP**-hard (under standard complexity theoretical assumptions). Yet no polynomial-time algorithm is known for this problem. A very recently announced [123] algorithm by Lackenby runs in quasipolynomial time.

KNOT EQUIVALENCE
Input: Two knots K_1 and K_2 described by knot diagrams of complexity n_1 and n_2.
Output: Are K_1 and K_2 isotopic?

Perhaps surprisingly, this problem seems (at least according to the current state of the art) much harder than UNKNOT RECOGNITION. It is known that KNOT EQUIVALENCE is decidable, and Coward and Lackenby [42] proved that any two diagrams of the same knot with respectively n_1 and n_2 crossings can be transformed one into the other using at most

$$2^{2^{\cdot^{\cdot^{\cdot^{2^{n_1+n_2}}}}}}$$

Reidemeister moves, where the height of the tower of exponentials is $k^{n_1+n_2}$, and $k = 10^{1000000}$. This directly gives an algorithm for KNOT EQUIVALENCE, albeit not a very efficient one. Kuperberg [116] proved that testing homeomorphism for closed orientable 3-manifolds is elementary recursive (which forbids towers of exponentials of unbounded height and is thus a stronger bound than the previous one), and it seems likely that the arguments can be generalized to the boundary case, which thus, coupled with the Gordon-Luecke theorem [79], would also imply that KNOT EQUIVALENCE is elementary recursive. No better algorithm seems to be known. Strikingly, no lower bound is known either: it is open whether this problem is **NP**-hard, and also whether it is at least as hard as GRAPH ISOMORPHISM (note that the closely related problem of testing whether two 3-manifolds are homeomorphic is known [121] to be at least as hard as GRAPH ISOMORPHISM).

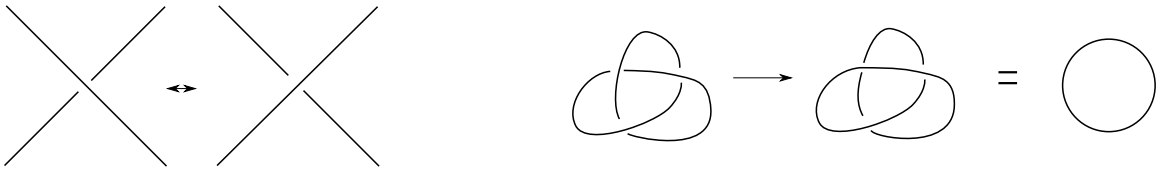

FIGURE 6.1 – Left: A crossing change. Right: A single crossing change transforms a trefoil knot into an unknot.

UNKNOTTING NUMBER
Input: A knot K described by a knot diagram of complexity n and an integer k.
Output: Is there a sequence of k crossing changes on K transforming it into a trivial knot?

Some precision is in order for the UNKNOTTING NUMBER problem. A crossing change is a change in the choice of which strands are above and below in a knot diagram, see Figure 6.1. When we ask for a sequence of crossing changes, these crossing changes are allowed to happen in *any* diagram of the knot. Furthermore, there are examples [16, 145] where the optimal sequence of crossing changes does not start from a diagram with a minimal number of crossings. This makes the UNKNOTTING NUMBER resilient to brute-force algorithms, since one would have to test an infinite number of diagrams of the same knot. In fact, it is unknown whether computing the unknotting number is decidable. Even testing whether a knot has unknotting number one is not known to be decidable. The closest result is a recent work of Lackenby [122], who provided an intricate algorithm to test whether a link with some mild hypotheses has splitting number [1] one. Here again, no lower bound is known, and the problem is not even known to be **NP**-hard.

The wide gaps between the best known lower bounds and upper bounds in these last two problems beg for a deeper inquiry of the computational complexity questions surrounding knot theory. With Yo'av Rieck, Eric Sedgwick and Martin Tancer, [E], we undertook this task and obtained **NP**-hardness proofs for three problems, yielding the following theorems.

Theorem 6.1.1. *[E, Theorem 2(a)] Given a link L_1 described by a link diagram and an integer k, the problem of finding a k-component sublink $L_2 \subseteq L_1$ that is trivial is **NP**-hard.*

The **NP**-hardness of finding a sublink of a fixed type had already been established by Lackenby [121], and our result strengthened it to trivial links. As we shall see, the proof of Theorem 6.1.1 is very simple and provides a good warm-up for the next two hardness reductions.

1. Similar to the unknotting number, the splitting number of a link is the minimal number of crossing changes (in any diagram) required to split a link, i.e., pull apart at least two of its components.

Theorem 6.1.2. *[E, Theorem 2(b)] Given a link L described by a link diagram and an integer k, the problem of deciding whether L has unlinking number has most k is **NP**-hard.*

Here, the unlinking number is the generalization of the unknotting number where we ask for k crossing changes in any diagram of the link L to yield the unlink. The use of links instead of knots is crucial for the analysis here, hence why we fall short of establishing **NP**-hardness for the unknotting number. As a byproduct, our proof also provides **NP**-hardness for a wide family of what we call intermediate invariants, which are related to unknotting numbers and surfaces embedded in four dimensions having the link as a boundary (see more details in Section 6.2).

Theorem 6.1.3. *[E, Theorem 1] Given a diagram K of a trivial knot and an integer k, the problem of deciding whether there exists a sequence of at most k Reidemeister moves transforming K into a trivial diagram is **NP**-complete.*

While, as we saw, detecting a trivial knot is not expected to be **NP**-hard, Theorem 6.1.3 suggests that finding the best way to actually untangle the knot using combinatorial moves is actually **NP**-hard. Of particular interest in this theorem is that it is a very rare occurrence of a hardness result for a problem pertaining to *classical* knots, i.e., knots embedded in $\mathbb{R}^3$. With the exception of the work on Kuperberg on approximating the Jones polynomial [115] and the work of Kuperberg and Samperton on coloring invariants [117], all the other known hardness proofs in 3 dimensions either involved links or different ambient manifolds. Arguably, the problem in Theorem 6.1.3 is a problem on knot diagrams and not knots, yet I believe Reidemeister moves to be sufficiently natural and well-studied for the problem to be considered a hard problem on classical knots.

The common point of Theorems 6.1.1, 6.1.2 and 6.1.3 is that in all three cases, the reductions follow a very similar pattern: we start from a 3-SAT instance, which we encode into a knot-theoretical problem using gadgets based on *Borromean rings*. Borromean rings are a 3-component link made of three unknots, with the property that removing any of the three components yields a trivial 2-component link. This property has a strong similarity with the 3-SAT clauses which are satisfied as soon as one of the literals is true, and this simple observation fuels all three of our hardness proofs. But before going into the technicalities, we first jump into a seemingly uncorrelated topic, where we nevertheless could make progress using similar techniques.

Let us note that, in an independent and simultaneous work, Koenig and Tsvietkova [111, 112] proved results similar to ours, and that our Theorem 6.1.3 answers one of their questions.

6.1.2 Embeddability questions and their computational complexity

Another class of hard topological problems, perhaps more intuitive to theoretical computer scientists, comes from embeddability questions. Here, we say that a k-dimensional simplicial complex K ***embeds*** into a topological space Y if there exists a proper injective

piecewise-linear[1] map $i : K \to Y$. The related computational problem, specialized to $Y = \mathbb{R}^d$, is the following.

> EMBED$_{k\to d}$
> **Input:** A k-dimensional simplicial complex K.
> **Output:** Does there exist a piecewise-linear embedding $i : K \hookrightarrow \mathbb{R}^d$?

For $k = 1$ and $Y = \mathbb{R}^2$, the embeddability amounts to the GRAPH PLANARITY problem which can be solved in optimal linear time using the Hopcroft-Tarjan algorithm [98]. Yet for other values of k, even in this "simple" case where $Y = \mathbb{R}^d$, testing embeddability is a delicate computational question, where the complexity depends in a subtle manner on the values of k and d. We refer to the introduction of Filakovskỳ, Wagner and Zhechev [68] (but see also [103, 171]) for a recent survey on these questions.

One of the reasons why the complexity of embeddability problems differs strongly depending on the ambient dimensions is because high dimension and/or co-dimension allows for topological tools (e.g., the Whitney trick) which allow to reduce the problem to a purely algebraic one. Yet in dimensions three and four, these tools are unavailable, giving the embeddability questions in these dimensions a very distinct flavor. In dimension four, not much is known, in particular the main open problem surrounding embeddability is whether EMBED$_{2\to 4}$ is decidable (it is known to be at least **NP**-hard [138]). In dimension three, the problems EMBED$_{2\to 3}$ and EMBED$_{3\to 3}$ were shown to be decidable by Matoušek, Sedgwick, Tancer and Wagner [137]. They left as an open problem whether the problem is **NP**-hard, hinting that the problem should be at least as hard as 3-sphere recognition, which is *not* expected to be **NP**-hard, since it is in **NP** [102, 167] and co-**NP** [185] (assuming the generalized Riemann hypothesis).

My first foray into this line of work was the following theorem, obtained with Benjamin Burton and Uli Wagner [L]:

Theorem 6.1.4 ([L])**.** *Given an integer g and a 3-manifold M, deciding whether the non-orientable surface of genus g embeds into M is **NP**-hard. If g is odd, the problem is **NP**-complete.*

This hardness proof suffered from two key drawbacks to be applied to EMBED$_{2\to 3}$: first the target manifold is not $\mathbb{R}^3$ but something more complicated, but even more, the manifold M is part of the input and changes as part of the reduction. Indeed, most of the hardness comes from the 3-manifold M getting more and more complicated. In contrast, in EMBED$_{2\to 3}$, the hardness has to come from the complex since the target manifold is fixed. The proof of **NP**-hardness in Theorem 6.1.4 follows a construction similar to the

1. We could also consider other categories, e.g., continuous or smooth maps. In dimensions two and three, the resulting embeddability problems are equivalent, but not in higher dimensions (see e.g. [138]).

NP-hardness of KNOT GENUS of Agol, Hass and Thurston [2], while the **NP**-membership relies on standard techniques in normal surface theory with a small amount of additional work.

A few years later, with Yo'av Rieck, Eric Sedgwick and Martin Tancer [H], we used a completely different approach to show that the problems $\text{EMBED}_{2\to 3}$ and $\text{EMBED}_{3\to 3}$ are indeed **NP**-hard:

Theorem 6.1.5. *[H, Theorem 1.1] The problems* $\text{EMBED}_{2\to 3}$ *and* $\text{EMBED}_{3\to 3}$ *are* ***NP****-hard.*

The proof of Theorem 6.1.5 relies heavily on knot theory. This might sound odd for the uninitiated since the embeddability problems do not really feature 1-dimensional embeddings. The connection is that the processes of *Dehn filling* and *Dehn surgery* allow to use a knot to transform a 3-manifold into another one. Using this formulation, a special instance of $\text{EMBED}_{3\to 3}$ asks whether a given 3-manifold with boundary can be Dehn filled along its toroidal boundaries to yield $\mathbb{R}^3$, and this is the problem that we prove to be **NP**-hard. The gadgets involved in this reduction rely on Borromean rings and are very similar to those used for the proofs of Theorems 6.1.1, 6.1.2 and 6.1.3.

Organization. The motivation of Theorem 6.1.4 via embeddability questions make it superseded by Theorem 6.1.5, and therefore in this chapter we will not discuss its proof, referring the reader to the corresponding article [L]. Then, while Theorem 6.1.5 came first in chronological order, it turns out that Theorems 6.1.1, 6.1.2 and 6.1.3 feature similar but simpler constructions and proofs, and thus it makes more sense to start with them in Section 6.2 and use them as a warm-up for the more intricate Theorem 6.1.5 in Section 6.3.

6.2 Borromean gadgets and hardness proofs

6.2.1 The trivial sublink problem

We start by explaining how to establish the **NP**-hardness of the following TRIVIAL SUBLINK problem:

> TRIVIAL SUBLINK
> **Input:** A link L described by a link diagram and an integer k.
> **Output:** Does L admit a trivial sublink with k components?

We first observe that there is a very easy reduction to prove **NP**-hardness of this problem from the INDEPENDENT SET problem, which is **NP**-hard [74]. I learned this reduction from Martin Tancer after we finished our project. Indeed, starting from an INDEPENDENT SET instance, one can simply replace each vertex with an octopus-shaped trivial knot in

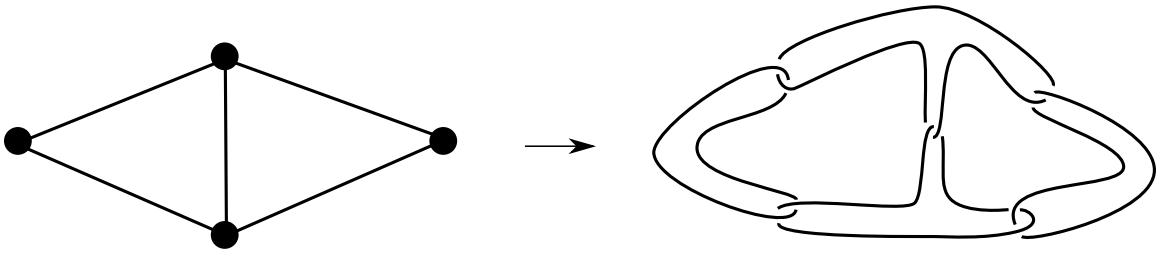

FIGURE 6.2 – Encoding INDEPENDENT SET in a TRIVIAL SUBLINK instance.

such a way that the knots corresponding to two adjacent vertices have linking number one, see Figure 6.2. Then a k-component sublink is trivial if and only if there exists a k-sized independent set.

However, the following different reduction will be more instructive. We start with a 3-SAT instance made of n variable and m clauses and embed it into a TRIVIAL SUBLINK in the following way:

1. Each pair of literals is represented as a Hopf link.

2. Each clause is represented by a Borromean link.

3. Then each literal is *banded* to its occurrences in the clauses, i.e. the corresponding unknots are connected via a small band, as in Figure 6.3. When a pair of such bands cross, one gets put under the other arbitrarily.

This reduces the 3-SAT problem into the TRIVIAL SUBLINK problem with $k = n$. Indeed, the proof that this works follows immediately from the key property of Borromean rings: as soon as one of the components gets removed, the link unravels and becomes an unlink. Indeed, if we have a satisfying assignment for our 3-SAT instance, we can remove from our link the corresponding literals. In each Hopf link, one of the components got removed, and in each Borromean ring, at least one component got removed, and therefore what remains is a trivial sublink made of n components. In the other direction, if we have a trivial sublink with n components, it means that we removed at least one component in each Hopf link, and at least one component in each Borromean ring. Since there are exactly n components in our trivial sublink and n variables, it means that exactly one component was removed from each Hopf link: this indicates our satisfying assignment.

6.2.2 Unlinking number and 4-dimensional problems

We now explain how almost the same reduction can be used to prove hardness for the following UNLINKING NUMBER problem:

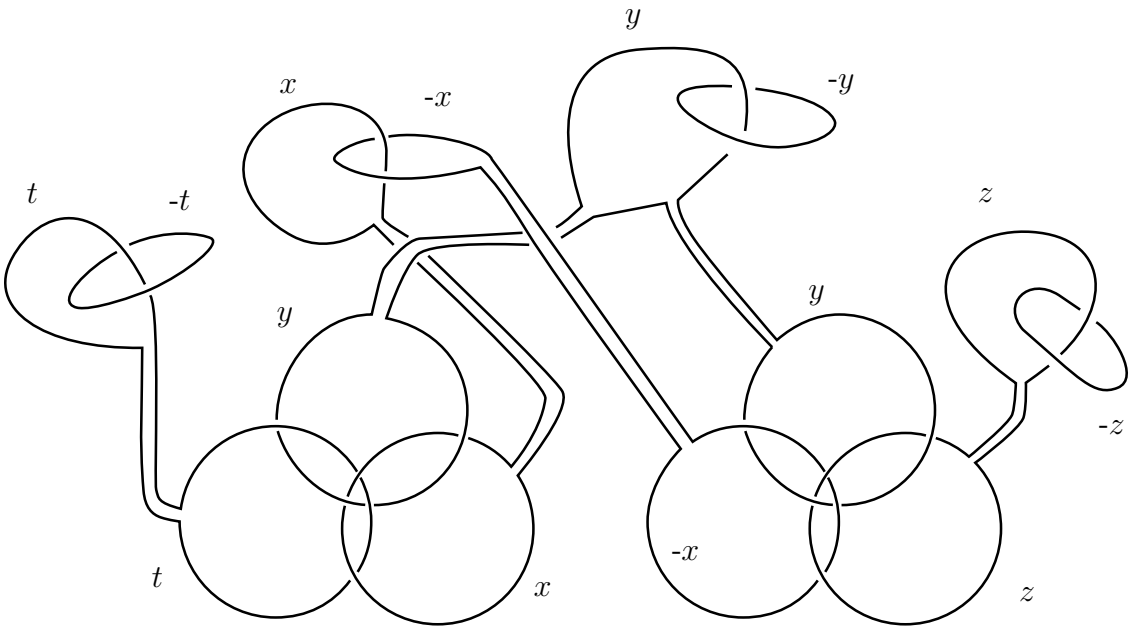

FIGURE 6.3 – The clause $t \vee y \vee x \wedge \text{-}x \vee y \vee z$ is encoded into a TRIVIAL SUBLINK instance.

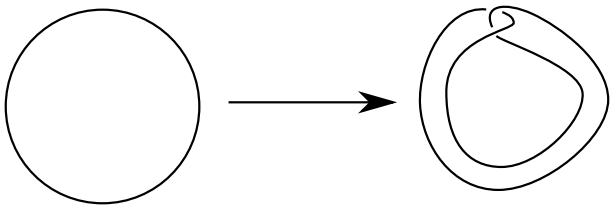

FIGURE 6.4 – The Whitehead double of a trivial knot.

UNLINKING NUMBER
Input: A link L described by a link diagram and an integer k.
Output: Can L be turned into a trivial link using at most k crossing changes?

The key construct involved here is the operation of ***Whitehead Doubling***, which consists in taking two parallel copies of a link component and closing them with a clasp in the manner [1] described by Figure 6.4. While it may seem rather innocuous at first glance, turning an unknot into another unknot, it does change the topological type when applied to

1. To be precise, we describe here the *untwisted* Whitehead double, defined by the fact that the two parallel copies have linking number zero. There are actually two ways of doing this, depending on the signs of the two added crossings, which are called the positive and the negative Whitehead double. We work with the positive one in this section.

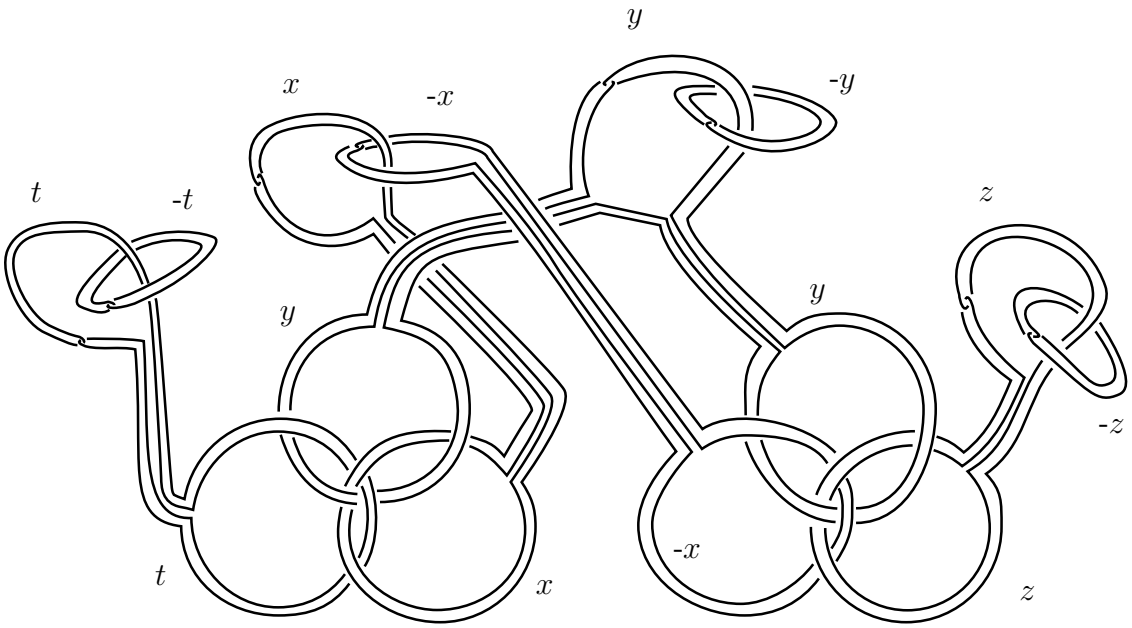

FIGURE 6.5 – The clause $t \vee y \vee x \wedge$ -$x \vee y \vee z$ is encoded into an UNLINKING NUMBER instance.

more complicated links made of multiple unknots. Furthermore, the connection with unlinking number is immediate: doing one crossing change at the location where the Whitehead double of an unknot is tied to itself allows to unravel the resulting link component, transforming it into an unknot isolated from the rest of the link. We call such a crossing change a ***standard*** crossing change.

The reduction to prove hardness of the UNLINKING NUMBER problem is obtained by taking the reduction for the TRIVIAL SUBLINK problem and replacing every link component with its Whitehead double, as pictured in Figure 6.5. The number k of crossing changes is taken to be the number of variables. One side of the proof that this reduction works is now immediate: if the SAT instance is satisfiable, we can look at the satisfied literals, and unravel each corresponding link components using the standard crossing change. After doing this, by the key property of the Borromean rings, everything in the diagram unravels and we have an unlink.

The reverse direction is trickier however, as we need to account for any possible crossing change, and not only the standard ones. However, we can still proceed with the proof using the following result:

Lemma 6.2.1 (Follows from [E, Lemma 18])**.** *The Whitehead double of the Hopf link and the Borromean rings are not unlinks.*

We use this result as follows. In each pair of literals x and -x, at least one crossing change has to happen since otherwise we do not have an unlink as per Lemma 6.2.1.

Since the allowed number of crossing changes is n, there is exactly one crossing change happening on the link components x and -x. This crossing change does not involve both components, since otherwise they would end up having linking number ± 1, and therefore they could not be the unlink. Therefore it only involves one of the components, and we choose the corresponding literal to be true. Now, in each of the Borromean rings, at least one of the link components must have had a crossing change, since otherwise we do not have an unlink as per Lemma 6.2.1. Therefore, the corresponding clause is satisfied by the assignment that we have defined, and this concludes the proof.

Let us make two remarks on this reduction:

1. In the previous argument, it is crucial that we have different link components to invoke linking numbers. Getting a hardness proof for the UNKNOTTING NUMBER problem requires therefore a different approach.

2. Unknotting and unlinking numbers can be interpreted topologically in the following way. The 3-sphere $\mathbb{S}^3$ can be thought of as the boundary of the 4-dimensional ball $\mathbb{B}^4$, and thus given a link in $\mathbb{S}^3$, one can ask whether there exists a smooth surface in this ball $\mathbb{B}^4$ having the link as a boundary, and what its genus is. It turns out that a sequence of crossing changes turning the link into an unlink can be interpreted as such a surface, with each crossing change contributing to the genus of the surface. Using this connection and a strengthened form of Lemma 6.2.1 due to Levine [127] allows us to also prove **NP**-hardness for a whole family of 4-dimensional invariants that we call intermediate invariants. We refer the interested reader to our paper [E] for more details.

6.2.3 Optimal number of Reidemeister moves

We now turn our attention to the problem of finding an optimal sequence of Reidemeister moves to untangle a diagram. We first introduce the problem for links, where hardness is easier to establish, and will then discuss how to turn this into a hard problem for knots.

> MINIMAL NUMBER OF REIDEMEISTER MOVES TO THE UNLINK
> **Input:** A link diagram L of the unlink and an integer k.
> **Output:** Can we transform L into the unlink using at most k Reidemeister moves?

Let us first ponder about how effective the various Reidemeister moves are at untangling a diagram. A Reidemeister move of type II- removes two crossings, a Reidemeister move of type I- removes a single crossing and a type III Reidemeister move removes none. So if we start with a diagram with n crossings and want to turn it into an unlink, there is a natural lower bound of n/2 moves, reached if and only if all the moves are type II-. From that perspective it makes sense to not count the number of Reidemeister moves in

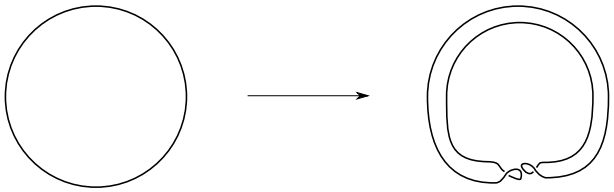

FIGURE 6.6 – The candy-wrapper gadget.

an untangling sequence, but to which extent they are slower than this (perhaps impossible) optimal sequence. The ***defect*** of a sequence of Reidemeister moves is twice the number of Reidemeister moves that we use minus the number of crossings in the initial diagram. Minimizing the defect is equivalent to minimizing the number of moves used, and the defect has the additional feature that it is invariant under Reidemeister moves of type II-.

The reduction to establish hardness for MINIMAL NUMBER OF REIDEMEISTER MOVES TO THE UNLINK also starts the same way as for the TRIVIAL SUBLINK problem. Here instead of replacing each link component with a Whitehead double, we replace it with the candy-wrapper gadget pictured in Figure 6.6. The interesting feature of this candy-wrapper is the following: by itself, it forms a trivial knot that one can untangle using a single type II- Reidemeister move. However, when it is crossed transversely by other strands, the best way to untangle it is to do a type I- Reidemeister move at each end, and then type II- moves all the way. Therefore, in the first case we have a sequence of moves of defect zero, while in the second case we have defect two.

This gadget allows us to encode the hardness of a satisfiability instance inside the order in which candy-wrappers are unmade, in the following way. Start with the reduction for TRIVIAL SUBLINK, and replace each link component with a candy-wrapper as in Figure 6.7. Assume that we have a satisfying assignment for a 3-SAT instance. Then we can untangle the corresponding diagram with defect $2n$, where n is the number of variables, as follows. For all the literals that are satisfied, we untangle the corresponding components, which costs us defect $2n$, as per the discussion in the previous paragraph. Then, as usual, in each Borromean ring, one of the components disappeared, and thus we can unravel those using only Reidemeister moves of type II-, contributing zero to the defect. Afterwards, we only see candy-wrappers with no transverse strands, which therefore can be untangled using only type II- Reidemeister moves. This completes the untangling sequence, using defect $2n$.

The reverse direction works similarly to the previous arguments. Assume that we have an untangling sequence with defect $2n$. Then in particular each pair of candy-wrappers for a variable must have been untangled. At least one move that is not of type II- must be spent for each of them (since there are no type II- moves available). Since the total budget for the defect is $2n$, we can show that the only possibility is that exactly two type I- moves are spent on each pair of literals involved in a variable. Such a move only involves one literal, which we can set to be true. Then, looking at the Borromean rings, there are also no type

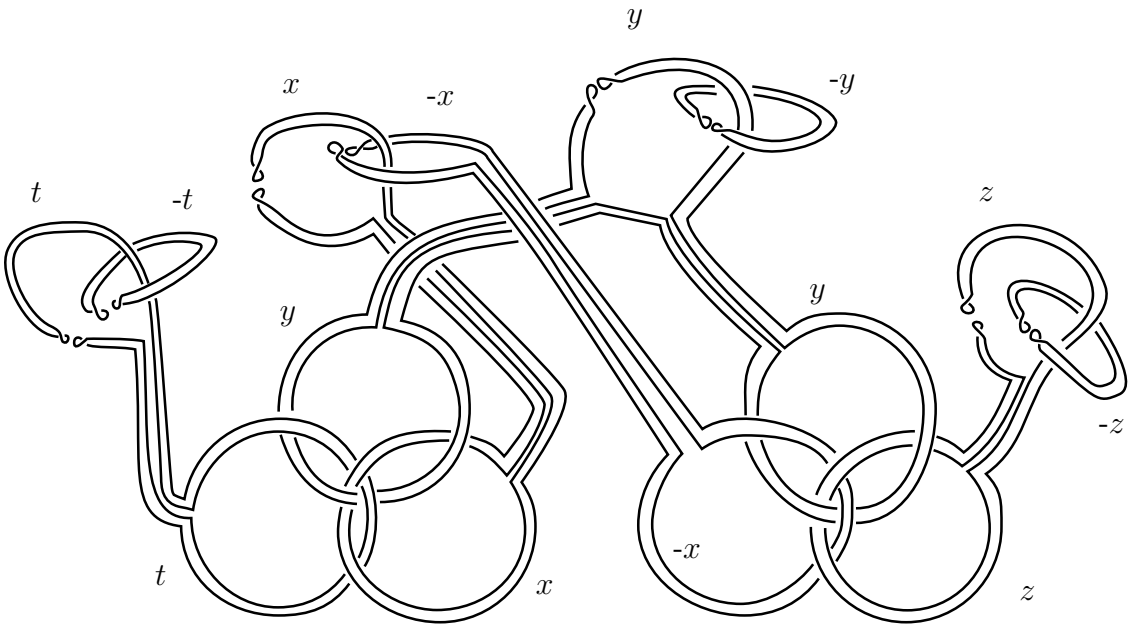

FIGURE 6.7 – A short sequence of Reidemeister moves to untangle this unlink provides a satisfying assignment for the clause $t \vee y \vee x \wedge \text{-}x \vee y \vee z$.

II- moves available, and thus one the link components must be untangled using a type I- move. This therefore shows that our assignment is a satisfying assignment for the 3-SAT instance, finishing the proof.

The same caveat as in the previous paragraph applies: in the argument, we relied crucially on analyzing link components separately, making the analysis hard to apply for the same problem with a single component, i.e., a knot:

> MINIMAL NUMBER OF REIDEMEISTER MOVES TO THE UNKNOT
> **Input:** A knot diagram L of the unknot and an integer k.
> **Output:** Can we transform L into the unknot using at most k Reidemeister moves?

Yet it is possible to use these gadgets to prove **NP**-hardness of MINIMAL NUMBER OF REIDEMEISTER MOVES TO THE UNKNOT. In order to do that, we start with the reduction for the link case, and attach everything to a common frame. Even the pictures start getting complicated, and we refer to [E, Part II, Figure 11] for an illustration.. Then the proof for one side of the reduction stays the same: given a satisfying assignment, we know in which order we should untangle the various gadgets for the literals in order to get to the unknot using defect n. The reverse direction however requires a delicate analysis, based on the local structure of the gadgets, in order to rule out the possibility that a strange sequence of Reidemeister moves outperforms the optimal one. We refer to [E, Part II] for the details.

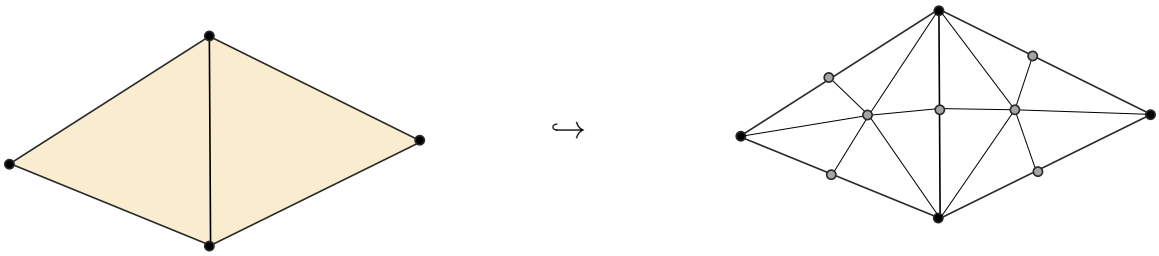

FIGURE 6.8 – We can embed the 2-dimensional faces (colored) in the triangles indicated by the barycentric subdivisions of the 1-skeleton (gray) (to simplify the picture, only one barycentric subdivision is realized here).

6.3 Hardness of embeddability into 3 dimensions

In this last section, we explain the connection between the hardness proof of the embeddability problem and the different gadgets that we introduced in the previous section.

Embedding 3-complexes via 2-complexes. The first step in the proof of Theorem 6.1.5 is to provide a polynomial-time reduction from $\text{EMBED}_{3\to3}$ to $\text{EMBED}_{2\to3}$. Therefore, proving **NP**-hardness for the former problem also establishes it for the latter problem. This reduction originates in the algorithm of Matoušek, Sedgwick, Tancer and Wagner [137, Section 12] and the main idea can be sketched as follows. Ideally, we would like to argue that a 3-dimensional complex embeds into $\mathbb{S}^3$ if and only if its 2-dimensional skeleton does. This is a bit too naive to be true: if we think about it one dimension lower, this is contradicted by the existence of planar graphs that admit non-planar embeddings. Yet if one rigidifies such a non-planar embedding by taking a few iterated subdivisions, then the 1-skeleton will no longer embed in the plane. Likewise, it turns out that embeddability of a 3-dimensional complex K amounts to embeddability of the 2-skeleton $K^{(2)}$ obtained by first cutting along all the cut-vertices, and then doing an iterated barycentric subdivision. Indeed, assume that we have embedded the 2-skeleton of this iterated barycentric subdivision. Then, the boundary of a tetrahedron of K is a 2-sphere, which has been embedded into $\mathbb{S}^3$. Topological embeddings of a 2-sphere into $\mathbb{R}^3$ can be wild (think of the Alexander horned sphere), but PL embeddings cannot: the PL Schoenflies theorem (see [137, Theorem 12.3] who cite Bing [14, Theorem XIV.1]) guarantees that this sphere is homeomorphic to a standard embedding of the 2-sphere in $\mathbb{S}^3$. Now we want to verify that one side of this sphere is empty, so that we can embed the tetrahedron τ inside: the idea is to choose the side where the iterated barycentric subdivision has been embedded, as it stands in the way of anything there being inside, except possibly components originating from cut-vertices, which have been treated separately. We refer to Figure 6.8 for an illustration one dimension lower.

Hardness of embeddings 3-complexes A particular subclass of 3-dimensional complexes is formed by 3-dimensional manifolds with boundaries. We can even restrict further and specify that the boundaries are tori. We will establish the hardness of deciding the embeddability of these very restricted 3-manifolds into $\mathbb{S}^3$.

A first step into understanding these embeddings is the following theorem of Fox, which we specialize here to toroidal boundaries:

Theorem 6.3.1. *[69] Let M be a 3-manifold whose boundary consists of a disjoint union of tori. Then if M embeds into $\mathbb{S}^3$, there exists such an embedding $f : M \to \mathbb{S}^3$ so that the complement $\mathbb{S}^3 \setminus f(M)$ consists of a union of solid tori.*

Therefore, if M embeds into $\mathbb{S}^3$, such an embedding can be obtained by plugging in solid tori in the "holes" of M. This suggests the following naive algorithm: do plug the holes, test whether the resulting manifold is $\mathbb{S}^3$ using one of the known algorithms, e.g., [167], and output the result. This fails because the plugging-in operation is ambiguous: attaching a solid torus T to a toroidal boundary ∂M requires an attachment map, i.e., a homeomorphism $f : \partial T \to \partial M$, and up to isotopy there are infinitely many of those homeomorphisms.

So we need to understand these homeomorphisms better. This forms the topic of *Dehn fillings*: we only give a hasty introduction here and refer to the article [H] or the book of Rolfsen [164, Chapter 9] for more background. One can see that the homeomorphism classes are parameterized by the homology class in ∂M of the image of a specific closed curve on ∂T called the meridian, which is the unique (up to homotopy) closed curve bounding a disk in T. Indeed, once one knows where to map this meridian, one can map the corresponding disk, and cutting the solid torus along this disk yields a 3-ball, which can be embedded in a unique orientation-preserving way (up to isotopy). Thus, this homology class in $H_1(\partial M)$ specifying where the meridian goes is of paramount importance: it is called a ***slope***. The operation of filling a boundary torus with a solid torus along a given slope is called a ***Dehn filling***. Reformulating with this language, our embeddability problem has now become the problem of deciding whether there exists a collection of slopes such that Dehn filling the boundary tori of M along these slopes yields $\mathbb{S}^3$.

The connection with knot theory is not yet apparent, and we explain it now. Given a knot K in a 3-manifold M, one can remove a tubular neighborhood $N(K)$ of K (which is a solid torus) and perform a Dehn filling on this new toroidal boundary, effectively plugging back a solid torus in place of the previous one. If the slope of the Dehn filling is different from the slope of the original solid torus, this might yield a different manifold, which we denote by M'. This operation is called ***Dehn surgery***, it provides an effective way to define a 3-manifold out of a classical knot and a slope (this combination is called a ***framed knot***). Furthermore, when starting with a classical knot K, there is a nice way to parameterize the homology group $H_1(\partial N(K))$: one can take a ***meridian*** (the only curve bounding a disk in $N(K)$), and a ***longitude***, the only curve bounding a surface in $\mathbb{S}^3 \setminus N(K)$. This provides us with a canonical identification of $H_1(\partial N(K))$ with $\mathbb{Z}^2$, and therefore a slope for Dehn

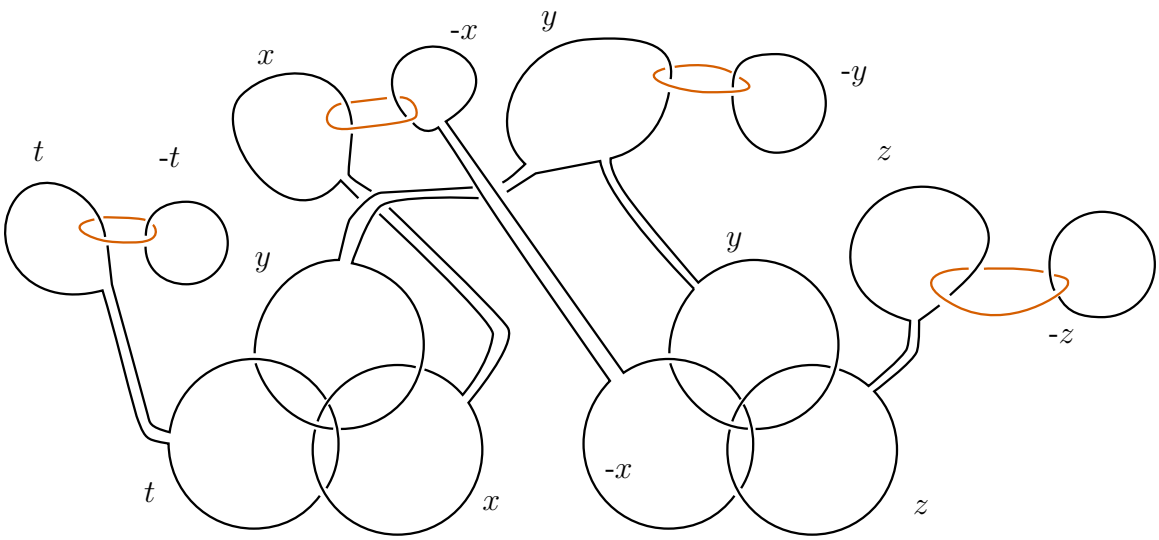

FIGURE 6.9 – Starting from a 3-SAT instance, we build a link as for the TRIVIAL SUBLINK problem, except that the literals are connected with *clasps*. Then the clasps (colored orange) are Dehn filled using a specific slope, and we dig out a tubular neighborhood of the other link components, yielding a 3-manifold with toroidal boundary.

filling can simply be described by a pair of integers (p, q). Furthermore, it is easy to check that a pair (p, q) corresponds to a simple closed curve if and only if p and q are relatively prime, and therefore it is standard to identify the set of slopes with the extended rationals $\frac{p}{q} \in \mathbb{Q} \cup \{\infty\}$.

Let us conclude this quick introduction with the following two facts:

1. Doing Dehn surgery on a knot K along a slope $1/0$ amounts to removing a solid torus and plugging it back exactly the way it was (since we map the meridian to the meridian), and thus doing nothing.

2. Let L be a Hopf link, and do Dehn surgery on one component with a slope $\frac{3}{2}$. Then if one does Dehn surgery on the other component with a slope $\frac{1}{1}$, the two surgeries cancel out and we obtain $\mathbb{S}^3$. This is one simple instance of the study of how different surgeries on link components interact, which is the topic of *Kirby calculus* [107].

We can now describe the prototype of our reduction. We will start from a 3-SAT instance and build a link out of it very similar to the TRIVIAL SUBLINK reduction: Borromean rings for the clauses, but instead of Hopf links for the literals we connect them with a *clasp*, as pictured in Figure 6.9. Then we will do Dehn surgery on the clasps with a well chosen slope (for example $3/2$, as above). For the other link components, we simply remove a tubular neighborhood and leave them unfilled, providing us with a 3-manifold with boundary M.

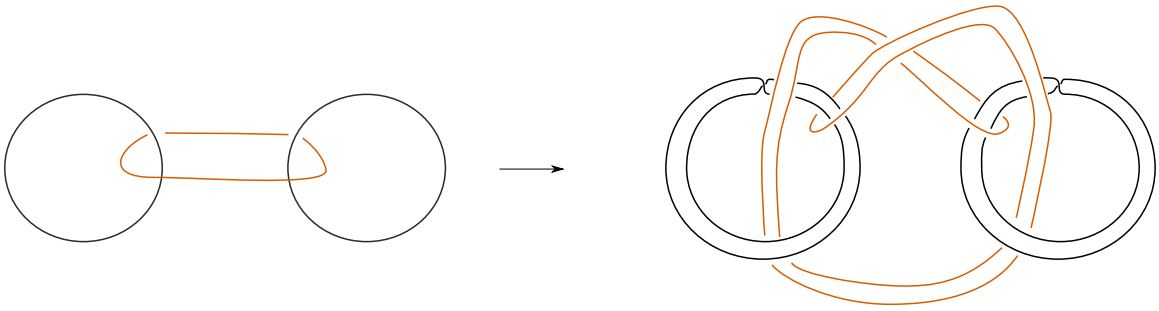

FIGURE 6.10 – Compared to the prototype described above, the actual reduction uses torus knots in place of unknots for the literals, and the clasp (in orange) that we use is more complicated.

The idea is now the following: the surgery that was done on each clasp has locally transformed the manifold into a more complicated one. In order to embed it into $\mathbb{R}^3$, we should use one of the two literals to undo this transformation. The other literal will simply be filled with a $1/0$ slope, which amounts to forgetting about that link component. Starting with a satisfying assignment for our 3-SAT instance, we can choose the true literal to be filled with a $1/0$ slope, and the false literal to be tasked with canceling the clasp (here with the slope $1/1$). Now, in each Borromean ring, one of the link components will thus have disappeared since one of the literals of the corresponding clause is satisfied. Therefore the whole picture unravels and we just have a family of Dehn surgeries on Hopf links, where the slopes have been carefully chosen to cancel each other. Thus we have embedded our 3-manifold into $\mathbb{S}^3$.

The reverse direction requires much more work. We want to argue that any family of Dehn fillings on M that yields $\mathbb{S}^3$ is of the form that we have just described, and thus that we can derive a satisfying assignment of variable for our 3-SAT instance from based on which literal is mapped to $\frac{1}{0}$. Unfortunately, having a precise understanding of which Dehn surgeries on links give which 3-manifold is a very intricate topic, home to some of the deepest theorems in twentieth-century knot theory, such as the Property R theorem proved by Gabai [72] or the Gordon-Luecke theorem [79]. While it seems plausible that the construction described above works and is a valid reduction, we fell short of proving it, and relied on a slightly different reduction, with the following two changes:

1. The clasp involved in the literals has a different shape, as in Figure 6.10. This allows us to leverage some results on the topology of the complementary space [100].

2. We use $(2, 1)$ torus knots instead[1] of unknots for the literals. This allows us to leverage results of Gordon and Litherland [80] on essential planar surfaces in cable spaces.

1. The attentive reader might notice that $(2, 1)$ torus knots are unknots! But once attached with the clasp, they will yield a different link than if we start with usual unknots.

Intuitively, by using more complicated gadgets, we rigidify the construction, making it more unlikely that Dehn fillings with the wrong slope "accidentally" yield $\mathbb{S}^3$. And indeed, we can prove it does not happen. The proof relies, in addition to the aforementioned ingredients, on an iterated application of a theorem of Culler, Gordon, Luecke and Schalen [44] relating the compressibility of surfaces before and after Dehn filling. We refer to the article [H] for the details.

CHAPTER 7

Other works on surfaces

7.1 Shortest path embeddings and crossing numbers

The material in this section comes from the article [M] which was co-authored with Alfredo Hubard, Vojtěch Kaluža and Martin Tancer.

A graph embedded on a surface can be concisely encoded using a combinatorial map, but this encoding is not very practical for representing, or visualizing the graph. When the graph is planar, an important literature has been devoted to the question of how to actually draw the graph so that its salient features are easily identifiable, a cornerstone being perhaps Tutte's "How to draw a graph" [179] where it is proved that for a planar triangulation, if we consider edges as springs, figuratively "nail" three vertices forming a triangle, and then let the system relax, we obtain a planar embedding of the graph. In particular this provides a proof of Fàry's theorem [66] that was mentioned at the end of Chapter 5, stipulating that a planar graph can always be embedded such that the edges are represented by straight lines. We refer to the Handbook of Graph Drawing [177] for an extensive study of the topic of drawing planar graphs.

Representing graphs embedded on surfaces is a much less studied topic. The aforementioned theorem of Tutte was generalized by Y. Colin de Verdière [39] in the following way: if one is given a surface S endowed with a Riemannian metric of non-positive curvature and a graph triangulating this surface, then, by considering the edges as springs and letting the system relax, one obtains an embedding of the graph where each edge is realized by a geodesic. But this theorem is much less useful from the perspective of visualizing an em-

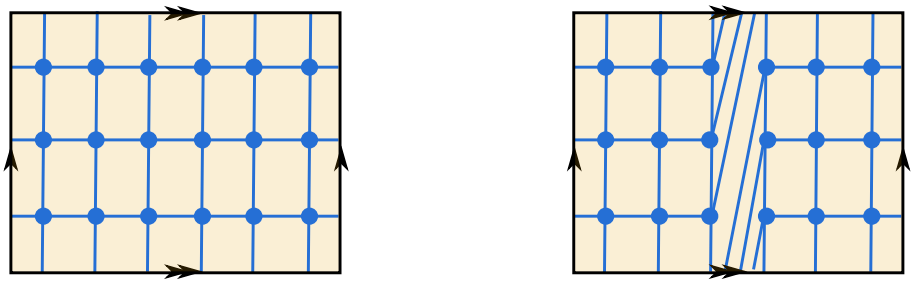

FIGURE 7.1 – Two grids embedded on a torus. While in both embeddings, all the edges are represented as geodesics, the second picture is clearly less informative than the first one.

bedded graph: even disregarding the numerical issues involved in the relaxation process, this requires visualizing the surface itself, for example in $\mathbb{R}^3$, and the geodesics might not provide a nice representation of the graph (see Figure 7.1). Another way to represent graphs on surfaces is provided by the Koebe-Andreev-Thurston circle-packing theorem [175]: the lesser-known version for graphs on surfaces shows that for any triangulated graph on a surface of genus at least two, there exists a hyperbolic metric on that surface such that the graph can be realized as the contact graph of a family of circles. In particular, in this realization, the edges of the embedded graphs can be realized as geodesics, and even better, as shortest paths. The drawback with this approach is that the metric depends on the graph, and therefore is not canonical.

Another approach, which relies less on geometric structures on the surface, is to follow the path laid out at the start of Chapter 5.3.4: an efficient way to manipulate a graph embedded on a surface is to compute a cut-graph, cut along it and investigate the resulting planar graph. But in contrast to the techniques presented in that chapter, in order to compare different graphs embedded on the same surface, it makes sense to enforce that the cut graphs used are identical. In this direction, the prototypical, and pretty much only result is a theorem of Lazarus, Pocchiola, Vegter and Verroust [125], which we formulate here in a cross-metric setting:

Theorem 7.1.1. *Let G be a graph of complexity n embedded cellularly on an orientable surface S of genus g. Then there exists a canonical system of loops for S that is transverse to G and has $O(gn)$ intersections with G.*

Here, a ***canonical system of loops*** is a family of loops based at one point forming an embedded graph with a single vertex and a single face, where the facial cycle is described by the word $a_1 b_1 \overline{a_1}\,\overline{b_1} \dots a_g b_g \overline{a_g}\,\overline{b_g}$. The canonical system of loops provided by Theorem 7.1.1 is computable in time $O(gn)$, and therefore, one can represent any graph embedded on a surface of genus g by first cutting along this system of loops and then representing the planar graph. The bound ensures that each edge does not get subdivided too many times during the cutting.

The proof of Theorem 7.1.1 heavily relies on the structure of the canonical system of loops and does not seem to easily generalize to any other cut graph that one might want to use. In particular, it is an open question whether the same upper bound holds for a ***non-orientable system of loops*** (a single vertex and single face graph embedded on a non-orientable surface with facial cycle $a_1a_1 \dots a_ga_g$), see Lazarus [124, Section 4.3]. While investigating the structure of flip graphs of triangulations of surfaces, Negami [146] encountered similar questions and conjectured that a similar bound should exist for any fixed cutting shape:

Conjecture 7.1.2. *Let G_1 and G_2 be two graphs embedded cellularly on a surface S of genus g. Then there exists a homeomorphism $h : S \to S$ so that G_1 and $h(G_2)$ intersect transversely $O(n_1n_2)$ times.*

Note that when choosing G_2 to be a canonical system of loops on an orientable surface, one recovers Theorem 7.1.1. Negami proved the weaker upper bound $O(gn_1n_2)$ in both the orientable and the non-orientable cases. Similar questions regarding upper bounds on crossing numbers have appeared in various works, for example in Matoušek, Sedgwick, Tancer and Wagner [136] in connection to embeddability questions and Geelen, Huynh and Richter [77] in connection to structural graph theory, and there is a related conjecture of Mohar [141]. Yet to my knowledge, no progress has been made on Negami's conjecture since its inception (see Archdeacon-Bonnington [5] and Richter-Salazar [158] for related results).

In a joint work with Alfredo Hubard, Vojtěch Kaluža and Martin Tancer, motivated by the above questions, we initiated the study of the following generalization of Fàry's theorem:

Problem 7.1.3. *Given a surface S, does there exist a Riemannian metric on S such that any simple graph G cellularly embeddable on S can be embedded such that edges are realized as shortest paths?*

Such a metric would constitute a best of both worlds scenario between the Y. Colin de Verdière theorem and the metric obtained by circle-packing. On the one hand we would get shortest paths and not geodesics, and on the other hand the metric would not depend on the graph one wants to embed. We called such a hypothetical metric a ***universal shortest path metric***. A positive answer to Problem 7.1.3 would immediately yield a proof of Conjecture 7.1.2, since shortest paths cross at most once. [1]

While we could not solve Problem 7.1.3 in full generality, and the main question is still open, we provided the following partial answers. In the following theorem, a k-universal

1. There is a slight discrepancy here: Problem 7.1.3 is stated for simple graphs as there is no hope of realizing loops as shortest paths, while Conjecture 7.1.2 is stated for more general graphs. One switches to the other setting by a constant number of subdivisions, which gets absorbed in the $O()$ constant of the conjecture.

shortest path metric is a metric such that any embeddable graph on a surface can be embedded with an embedding in which edges are drawn as concatenations of k shortest paths.

Theorem 7.1.4 ([M]). *1. The sphere, the projective plane, the torus and the Klein bottle can be endowed with a universal shortest path metric.*

2. *The standard square metric on the Klein bottle is not a universal shortest path metric.*

3. *On an orientable surface of genus g, for any $\varepsilon > 0$, with probability tending to 1 as g goes to infinity, a random hyperbolic metric is not a $O(g^{1/3-\varepsilon})$-universal shortest path metric. In particular, with probability tending to 1 as g goes to infinity, a random hyperbolic metric is not a universal shortest path metric.*

4. *For every $g > 1$, there exists an $O(g)$-universal shortest path hyperbolic metric on the orientable surface of genus g.*

An interesting aspect of Theorem 7.1.4 is that its proof leverages results from many different areas. The first item is obtained via a reduction to irreducible triangulations, which are in finite and small number for the relevant surfaces. The second item uses basic arguments from covering space theory. The third item builds on results on random hyperbolic metrics and Teichmüller theory from Mirzakhani [139] and Guth, Parlier and Young [86]: here, a random hyperbolic metric denotes a metric chosen uniformly at random for the so-called Weil-Petersson volume on the space of such metrics (see Farb and Margalit [65] for an introduction to Teichmüller theory). The last result follows from an octagonal decomposition devised by É. Colin de Verdière and Erickson [38] which we couple with the aforementioned theorem of Y. Colin de Verdière [39]. We refer to the article [M] for more background, details and proofs.

7.2 Homotopy moves

The material in this section comes from the articles [A] and [I] which were co-authored with Hsien-Chih Chang, Jeff Erickson, David Letscher, Saul Schleimer, Eric Sedgwick, Stephan Tillman and Dylan Thurston.

In various applied and theoretical settings, a question that comes up naturally is how to efficiently ***tighten*** a closed curve or a path. For example, given a set of obstacles in the plane and a path with fixed endpoints avoiding these obstacles, the classical SHORTEST

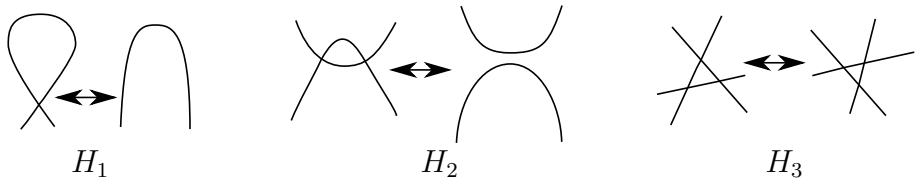

FIGURE 7.2 – The three homotopy moves.

HOMOTOPIC PATH problem asks to compute the shortest path homotopic to the input path in the space $\mathbb{R}^2 \setminus$ obstacles, and there are efficient algorithms for this problem [12, 25, 97]. A generalization of this problem was studied by É. Colin de Verdière and Erickson [38], where they provided efficient algorithms to compute a shortest homotopic path or closed curve on a combinatorial (or a cross-metric) surface.

In this chapter, we investigate a similar problem related to tightening a curve on a surface, but we are trying to control the number of elementary *combinatorial moves* involved in such a tightening. The setup is the following: we fix a surface S, and the input is a non-simple closed curve γ on the surface S. Putting γ in ***minimal position*** means finding another curve γ' which is homotopic to γ and with a minimal number of self-crossings. Similarly to the Reidemeister theorem for knots that we mentioned in Chapter 3, transversality arguments show that any homotopy can be realized by a finite sequence of ***homotopy moves***, which are pictured in Figure 7.2 (these are the same as Reidemeister moves without the decorations at vertices) and isotopies. Then the following questions come naturally:

1. Can we compute a minimal position of a curve? How efficiently?
2. How many homotopy moves are are needed to bring a curve to a minimal position?
3. Can a curve be put in minimal position using a ***monotonic*** sequence of homotopy moves, i.e., such that that the number of intersections never increases? If yes, how many moves are needed?

Furthermore, the above questions can also be asked for the more general setting of a ***multi-curve***, i.e., a family of closed curves, for which the minimal position is a family of closed curves homotopic to the input ones with a minimal number of intersections. An answer to the first question was provided by Despré and Lazarus [55], who provided a polynomial-time algorithm for the case of a single closed curve. An answer to the second question is provided by my first result on this topic, in a joint work with seven co-authors:

Theorem 7.2.1. *[I] On an orientable surface with or without boundary, a closed curve with n self-intersections can be tightened using $O(n^4)$ homotopy moves.*

Before that theorem, the only known bound was exponential, following from the work of de Graaf and Schrijver [48] or Hass and Scott [94] and an exponential bound on the number of surface maps [11]. We also provided a lower bound: a closed curve on an annulus

requiring $\Omega(n^2)$ homotopy moves to be simplified. The proof of Theorem 7.2.1 builds on a strong structural result of Hass and Scott [93] on the self-intersections of closed curves on surfaces, and involves repeatedly identifying so-called *singular bigons* and providing a way to simplify those with a controlled number of moves. This requires some intricate bookkeeping: while bigons in general are easy to simplify using homotopy moves, here the bigons are degenerate enough that we need to be very careful of not making one more complicated during the process of undoing it.

The answer to the third question is positive, and follows from independent work of de Graaf and Schrijver [48] and Hass and Scott [94]. In a follow-up paper with Hsien-Chih Chang [A], we refined these techniques to provide a polynomial upper bound for the third question, and at the same time generalize the results for the other questions to the more general setting of multicurves.

Theorem 7.2.2 ([A])**.**
- *On an orientable surface of genus g with non-empty boundary, any multi-curve with n self-intersections can be tightened monotonically using $O((g+b)n^3)$ homotopy moves.*
- *On an orientable surface without boundary that is not a torus, any multi-curve with n self-intersections can be tightened monotonically using $O(n^5 \log^3 g/g^2 + gn^3)$ homotopy moves.*

Furthermore, the moves can be computed efficiently, thus also yielding the first algorithms to put a multi-curve in minimal position in polynomial time. It is a striking fact that this theorem does not apply to the torus, as problems on tori tend generally to be easier than on surfaces of higher genus. The reason here is that our arguments leverage hyperbolic geometry, which has distinctive features which are lacking for the Euclidean geometry that a torus can naturally be endowed with. A second tool that we rely on to prove Theorem 7.2.2 is a combinatorial model that we call a *pipe system* and that gets iteratively refined as the tightening process happens. It originates from recent algorithms to draw planar graphs with restrictions [35, 40, 71].

Let us now provide some additional motivation for Theorem 7.2.2 by explaining its connections to ***electrical transformations*** on (embedded) graphs. Given G an embedded graph on a surface S, the set of electrical moves pictured in Figure 7.3 provides rules to transform G into another graph [1]. The origin of these moves (and of their names) is that they provide simplification rules to compute the equivalent resistances of resistance networks (see for example [3]), but these moves also have numerous applications in other fields. For example they are used in Steinitz's original proof of his theorem that 3-connected planar graphs coincide with the graphs of three-dimensional convex polyhedra [174], or they are prominently featured in the structural theory of linkless graphs [160]. In many

1. We focus in this chapter on *embedded* electrical moves, i.e., we insist that faces pictured in Figure 7.3 be empty for the move to be allowed. The more general theory is more subtle, we refer to the thesis of Chang [34] for more details.

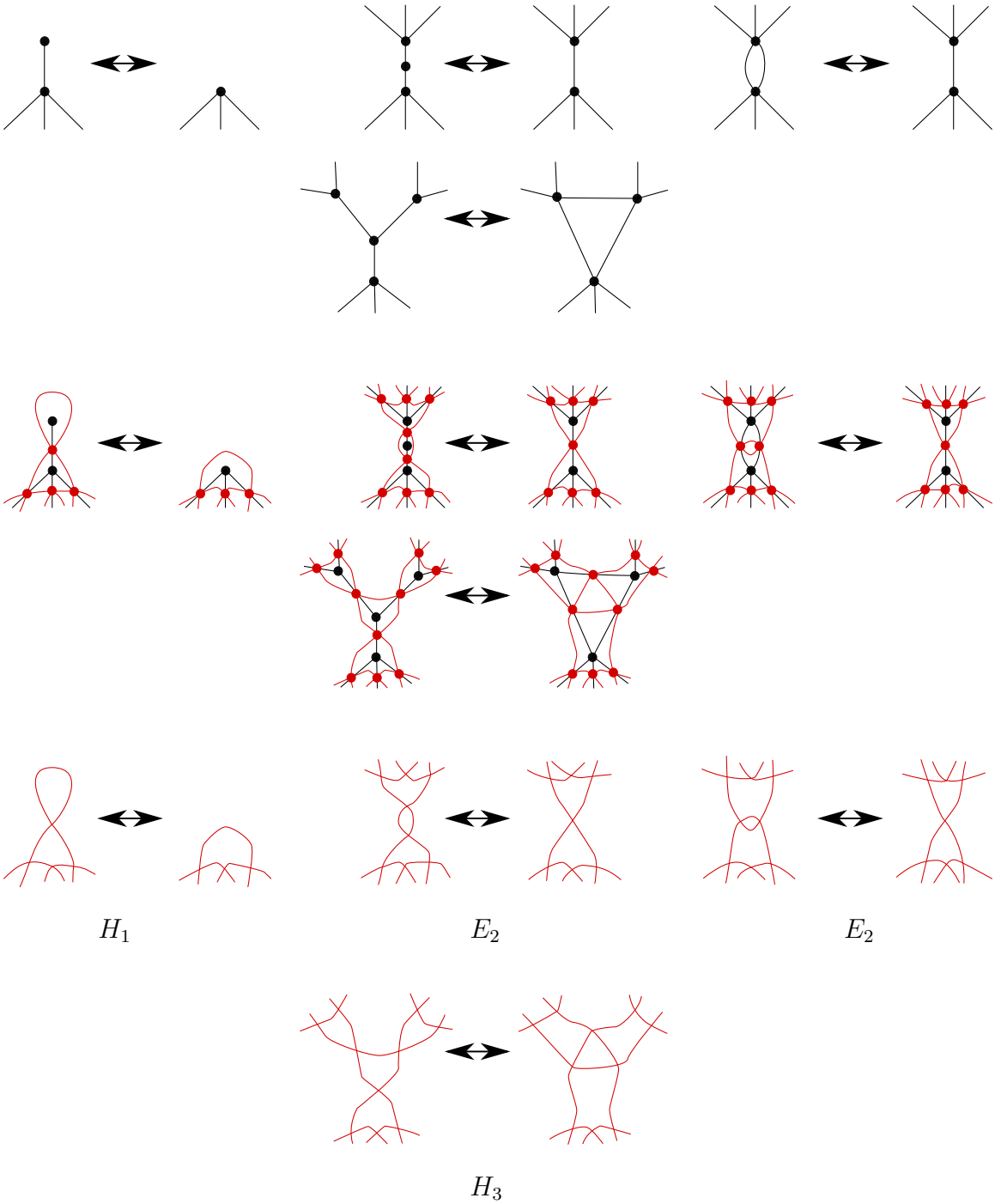

FIGURE 7.3 – The four electrical moves (top), their effect on the medial graph (middle) and their similarity to homotopy moves (bottom)

applications, the goal is to simplify as much as possible a graph using these transformations, i.e., transform the graph into a graph with a minimal number of vertices.

The connection between electrical transformations and homotopy moves operates via the ***medial graph***, which obtained from an embedded graph by placing a vertex on each edge, and connecting two vertices placed on adjacent edges, where two edges are considered adjacent if they occur consecutively on a facial cycle. Note that a medial graph has all its vertices of degree 4, and thus can be interpreted as a multicurve. As pictured in Figure 7.3, electrical transformations *almost* amount to homotopy moves on this multicurve. The key difference is that the second homotopy move is different, instead of an H_2 move we get what we denote by an E_2 move, which yields interesting difficulties.

Our Theorem 7.2.2 can be applied to the problem of simplifying an embedded graph using electrical transformations as much as possible as follows. We first compute the medial graph. Then we compute a monotonic sequence of homotopy moves to tighten these curves, and apply them, until we encounter an $H2$ move. At this stage, we apply the $E2$ move instead. This completely rearranges the multicurve and their homotopy classes, but we can start the process again: compute a new monotonic sequence of homotopy moves, and apply them until we encounter an $H2$ move, etc. Since the $E2$ moves reduces the number of self-intersections by 1, this process only lasts for a polynomial time, and thus we obtain a polynomial sequence of electrical transformation moves to simplify our graph. At this stage, two remarks are in order:

- It is not clear that we are done. It could be that we obtain an embedded graph, and the corresponding medial multicurve is in minimal position from the perspective of homotopy moves, but that applying a well-chosen $E2$-move in reverse shuffles everything and provides a way for further simplification. It is a delicate theorem of Chang, Cossarini and Erickson [33] that this actually does not happen.

- For this argument to work, it is crucial that we work with a *monotonic* sequence of homotopy moves, and thus that we have at our disposition Theorem 7.2.2 instead of just Theorem 7.2.1. Indeed, in a non-monotonic sequence, a move $H2$ increasing the number of vertices would have to be exchanged with an increasing $E2$, completely shuffling the topology of the curves, and there would be no way of controlling the resulting number of moves.

Thus, the previous discussion yields the following corollary.

Corollary 7.2.3. *[A] Any graph embedded on a surface can be reduced as much as possible using electrical transformations in polynomial time.*

This result also applies in a setting with *terminals*, i.e., vertices that have to be left invariant by the electrical transformations, although this involves some subtleties regarding *terminal-leaf reductions* that we will not discuss here (we refer to [A, Section 5.3]). Before our work, the result of Corollary 7.2.3 was only known in the setting of planar graphs with up to four terminals [53].

CHAPTER 8

Other works on knots and 3-manifolds

8.1 Treewidth of knots

The material in this section comes from the article [F] which was co-authored with Jessica Purcell, Saul Schleimer and Eric Sedgwick.

As we surveyed in Chapter 6, computational problems involving knots are generally hard, or at least they seem to be in the sense that we do not know efficient algorithms to solve them. In the past decades, a vast literature in theoretical computer science was devoted to developing new angles of attacks to solve hard problems, primarily focusing on two directions: approximation algorithms (see for example Vazirani [180]) on the one hand, and parameterized algorithms (see for example Cygan et al. [45]) on the other hand. This section is devoted to the latter approach. A widely beloved parameter enabling tractability of hard problems is the *treewidth* of a graph: in intuitive terms, it measures quantitatively how close to a tree a graph is. This parameter is very relevant for algorithmic purposes since most algorithmic problems are easy to solve on trees (generally using some form of dynamic programming), and therefore can also be solved efficiently for graphs that "look like trees". More precisely, many **NP**-hard problems are FPT when parameterized by the treewidth, i.e., can be solved on graphs of treewidth at most k in time $f(k)poly(n)$, where f is some computable function.

Parameterized algorithms can be applied to problems in knot theory via knot diagrams. Indeed, a knot diagram, once one forgets the information of which strand goes over or under at a crossing, is merely a 4-valent planar graph, with a well-defined treewidth, and thus

one can hope that standard knot-theoretical questions become easier under the hypothesis that this treewidth is small. Unfortunately, even this is unknown for the most important questions: in particular it is unknown whether UNKNOT RECOGNITION can be solved in polynomial time when the input diagram has fixed treewidth (the best known result is such an algorithm for inputs of treewidth 2 [17]). Yet this approach is known to be successful for a host of easier questions, notably for the problem of computing a wide family of knot invariants, such as the Jones polynomial [130], the HOMFLY-PT polynomial [22] and more refined versions of those [131]. Note that a well-known conjecture posits that the Jones polynomial detects the unknot, which thus would yield a positive answer to the FPT status of UNKNOT RECOGNITION parameterized by the treewidth.

A question that immediately arises from these endeavors is to understand which knots admit knot diagrams of small treewidth. The question was asked in the reverse direction by Burton [21]: does there exist a constant c such that every knot K admits a diagram of treewidth at most c? Makowsky and Mariño asked a similar question in [130]. The motivation is clear: if the answer is positive, and furthermore such a small-treewidth diagram can be computed efficiently, then the previous parameterized algorithms provide a very plausible approach towards solving UNKNOT RECOGNITION in polynomial time. Of course, the consensus was that the answer should be negative, but while it is easy to build diagrams of a given knot with arbitrarily high treewidth, proving the nonexistence of small treewidth diagrams for a given knot seems much harder.

In a joint work with Jessica Purcell, Saul Schleimer and Eric Sedgwick, we resolved the question in the negative. A ***torus knot*** $T_{p,q}$ is a knot defined by taking the standard embedding of a torus as a torus of revolution in $\mathbb{R}^3$, and considering a curve winding p times around a circle in the interior of the torus and q times around the axis of symmetry.

Theorem 8.1.1. *[F] Any diagram of the torus knot $T(k, k+1)$ has treewidth $\Omega(k)$.*

Our theorem is actually more general, and provides topological criteria for a knot to not admit a diagram of low treewidth, in terms of its *bridge number* and the existence of *incompressible planar surfaces* in its complement, we refer to the article [F] for more details.

The idea of the proof is to connect the very combinatorial notion of treewidth to more topological notions that have been heavily investigated in knot theory. An impressionistic description is as follows: we assume by contradiction that a torus knot $T(k, k+1)$ admits a diagram of low treewidth and reach a contradiction. Famous work of Seymour and Thomas [170] shows that a planar graph of low treewidth can be interpreted geometrically as a graph which can be swept by disjoint simple closed curves such that each of these closed curves intersects the graph a bounded number of times, and the family that they form nests in the shape of a tree. When the input graph is a knot diagram, we show that these closed curves can be lifted to spheres in $\mathbb{R}^3$, yielding what we call a *sphere decomposition*: a family of disjoint spheres in $\mathbb{R}^3$, each of them having small intersection with the knot K, nested in a tree-like fashion, and such that the layout of the knot with respect to these

spheres is very constrained. The last ingredient is to show that such a sphere decomposition is very closely related to the classical notion of a *Heegaard splitting* of a 3-manifold. More precisely, we show how to upgrade a sphere decomposition into a *multiple Heegaard splitting* (see Hayashi and Shimokawa [96]) for the knot $T(k, k+1)$ in $\mathbb{R}^3$, and the low treewidth hypothesis implies that the *complexity* of this splitting has to be low. But leveraging the work of Hayashi and Shimokawa [96], we show that such a low-width splitting would prove the existence of some particular surfaces in the knot complement, which are known not to exist for torus knots (and other families of knots), concluding the proof.

8.2 Hardness of the link crossing number

The material in this section comes from the article [B] which was co-authored with Marcus Schaefer and Eric Sedgwick.

The ***crossing number*** of a knot or a link is the minimum number of intersections in a diagram of the knot or link. While it is arguably the most natural numerical invariants to consider for knots, and indeed most knot tables classify those by their crossing number [23], it is also famously intricate, as it is hard to relate the crossing number to topological properties of the knots. For example, it is a famous open question whether $cr(K_1 \# K_2) = cr(K_1) + cr(K_2)$, where $\#$ denotes the connected sum of knots (see Lackenby [118] for the best known bounds on this question). However, the discovery of the Jones polynomial in the 80s led to renewed interest in the crossing number and more generally in a diagrammatic approach to knot theory, in particular because it led to easy proofs of the famous Tait conjectures (we refer for example to the book of Kaufmann [104]).

With Marcus Schaefer and Eric Sedgwick [B], we undertook the study of the crossing number of knots and links from a computational perspective. Our path here was guided by the analogous notion in graph theory: the ***crossing number*** of a graph G is the minimum number of crossings of a drawing of G in the plane. The crossing number of graphs is also a notoriously delicate quantity. For example, computing it is **NP**-hard [75], and the value of the crossing number of complete or complete bipartite graphs is famously unknown [165]. This suggests trying to encode a graph crossing number in an instance of computing the crossing number of a knot or a link, which we succeeded in doing, establishing the following theorem.

Theorem 8.2.1. *[B] Let L be a link described by a sequence of segments (or a diagram) and k be an integer. Deciding whether L has crossing number at most k is **NP**-hard.*

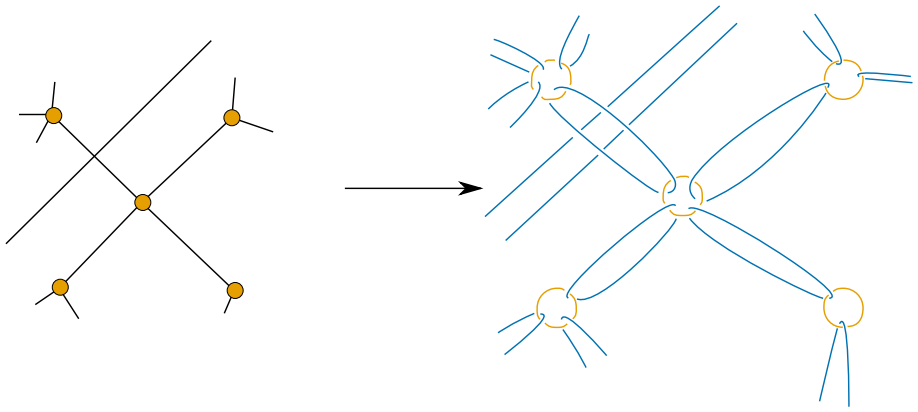

FIGURE 8.1 – A naive reduction from graphs to links.

Let us observe that this is far from telling a complete story on the complexity of computing the crossing numbers of knots and links. Indeed, the best algorithm known for this problem seems to be to try all the possible diagrams up to k crossings, and for each of them test whether it is equivalent to the input link, using the galactic algorithms mentioned in Chapter 6. Bridging the gap between the lower and upper bound seems like a formidable open problem. Furthermore, Theorem 8.2.1 only applies to links, for the same reason as most of our reductions in Chapter 6: links provide more rigidity and control than knots, making it easier to devise reductions. Proving an analogue of Theorem 8.2.1 for knots, or even for links with a fixed number of connected components, seems out of reach of our techniques.

The proof of Theorem 8.2.1 follows an expected pattern until one encounters an unexpected difficulty, which I would like to explain here, leaving the interested reader to check its resolution in the article [B]. We start from an instance of the graph crossing number problem that is known to be **NP**-hard and try to transform it into a link, so that the crossings of the link necessarily mirror the crossings of the graph. A naive version of this reduction is pictured in Figure 8.1, where we see a first difficulty: the topological type of the link that we obtain will depend on the rotation scheme of the starting graph. Therefore, we want to start from an instance of the graph crossing number that is **NP**-hard even when fixing the rotation scheme, we chose one due to Muñoz, Unger and Vrt'o [144].

Then, we want to argue that a diagram of the link with a minimal number of crossings corresponds to a drawing of the original graph. This is a matter of *rigidity*: the goal is to show that the flexibility in diagrams of the link exactly mirrors the flexibility of the graph drawing. In order to enforce this rigidity, an important trick that is used in many graph-theoretical reductions on crossing numbers is to put weights on edges, which can easily be simulated using multiple edges (if these are forbidden, one can subdivide those). Indeed, in a drawing of a graph of minimal crossing number, one can always arrange parallel edges so that they are drawn just next to each other, since drawing those in a small neighborhood

of the one with a minimal number of crossings does not increase the crossing number. Yet interestingly, this trick is hard to apply in the context of links, as a component might self-intersect in the optimal diagram, and copies of self-intersecting components yield a crossing number growing quadratically with the number of components. Basically, we were unable to argue that in a crossing-minimal drawing, all copies of a knot would have to be drawn the same way. In order to circumvent this issue, we still rely on a weighting scheme, but change the equivalence notion that we use for knots to argue about the weighting, before ultimately connecting back this new equivalence notion with the classical one.

CHAPTER 9

Perspectives

We conclude this habilitation thesis with some perspectives on ongoing and future research directions. We start by highlighting ongoing work with PhD and Masters students, and then outline further research topics. In line with the organization of this document, we place a specific emphasis on the open problems stemming from the three lines of research presented in Chapters 4, 5 and 6.

9.1 Work with students

I have worked on numerous occasions with PhD students, which are identified with stars in my list of publications. In particular, the articles [A, I] and [C, G, J] are the results of long collaborations respectively with Hsien-Chih Chang and Tim Ophelders during their PhD theses.

I am currently co-supervising three PhD students, and for two of them I have also (co-) supervised their master theses.

With Niloufar Fuladi, a PhD student co-supervised with Éric Colin de Verdière and Alfredo Hubard, we investigate topics in the continuation of the results on intersections of embedded graphs and shortest path embeddings presented in Chapter 7. As mentioned in that chapter, the directions are multiple as most questions one can ask remain wide open. Our first focus is on non-orientable surfaces, which were a blind spot of our initial work [M]. The following preliminary results have already been obtained during Niloufar Fuladi's master thesis: first, there is a subtle flaw in Negami's proof that two graphs embeddable on a non-orientable surface of genus g can be reembedded so as to cross $O(gn_1n_2)$ times, which we corrected using a different argument. Second, we were able to generalize the fourth item of Theorem 7.1.4 to the setting of non-orientable surfaces by providing a new decomposition scheme for those, akin to the octagonal decomposition defined by

É. Colin de Verdière and Erickson [38] for orientable surfaces. The next natural direction, topic of ongoing work that is nearing conclusion, is to obtain an analogue of Theorem 7.1.1 for non-orientable surfaces and non-orientable canonical systems of loops. Here, the main starting point is a recent result of Schaefer and Stefankovič [166] providing very strong bounds on how to embed a non-orientable map with a geometrized model of crosscaps.

With Jean Chartier, a PhD student co-supervised with Laurent Hauswirth and Stéphane Sabourau, we investigate questions related to min-max techniques akin to those presented in Chapter 5 on homotopy height. As we explained in that chapter, optimal homotopies and their higher dimensional generalizations (sweep-outs) are widely studied in minimal surface theory as they provide a powerful tool to prove the existence of geodesics, or more generally minimal surfaces. These techniques have been under the spotlight in recent years, leading to the resolution of longstanding conjectures such as the Willmore conjecture [133] or Yau's conjecture [173]. Yet many open questions remain, even in two dimensions. We pursue two directions. First, while geodesics can be proved to exist using minmax techniques, no tools seem to exist to compute those, even in metric structures which should be amenable to computation. For instance, a theorem of Pogorelov [153] proves that there always exist at least three distinct simple *quasi-geodesics* [1] on convex polyhedra, yet it is an open problem to compute such a quasi-geodesic in polynomial time. As mentioned in Demaine, Hertserberg and Ku [52], even the existence of an algorithm is not clear, since there is no obvious bound on the number of edges that these geodesics intersect. We are investigating how minmax techniques can be made algorithmic to actually compute these quasi-geodesics. In a different direction, an open question of Hass and Morgan [92] asks whether any Riemannian 2-sphere possesses a geodesic *net* with three faces, i.e. an embedded graph with three faces where each edge is a geodesic and the angles at each vertex are balanced (i.e., the sum of vectors of norm 1 tangent to each incident edge is 0). They proved the existence of such a net when the manifold has positive curvature, but the argument does not generalize to a curvature-free setting. Our aim is to develop min-max techniques to capture the existence of such objects.

With Corentin Lunel, a masters student who is now starting a PhD thesis co-supervised with Éric Colin de Verdière and Pierre Dehornoy, we pursue the work described in Chapter 8 on treewidth of knots. The main result of our article [F] requires quite a bit of topological background, in the form of the reliance on Hayashi-Shimokawa [96], as it aims at proving the *non-existence* of some topological sweep-out of the knot, and proving non-existence is often harder than proving existence. Structural graph theory provides tools to alleviate this difficulty, in the form of *obstructions* to small decompositions. For example, such an obstruction for branch-width is a *tangle* (completely unrelated to the knot-theoretical concept bearing the same name), and it works like this: a graph has branch-width at least k if and only if it admits a tangle of size k. This turns the problem of lower-bounding the

1. A quasi-geodesic is a geodesic except at the singular points of the polyhedron, where it is required to form an angle at most π on both sides.

branchwidth upside down, instead of proving the non-existence of a specific branch decomposition, we "just" have to prove the existence of a tangle. The main result of the master thesis is to properly define a topological version of this concept of tangle in order to recover Theorem 8.1.1. The technical aspects of this construction build on an unexpected connection with the tools used by Pardon [149] to bound from below another invariant called the *distortion* of the knot. During his PhD thesis, we aim at developing our understanding of this connection between structural graph theory and knot theory to develop new algorithmic and combinatorial tools surrounding small-width decompositions of knots.

9.2 Further research directions

Algorithms for graphs on surfaces. In Chapter 4, we saw that topological ideas and algorithms can be crucial when developing algorithms for graphs embedded on surfaces: our approximation algorithm for the very combinatorial in nature MULTICUT problem relies under the hood heavily on topological primitives such as covering spaces and homotopic shortest paths or cycles. A natural extension of this chapter is to develop exact or approximate algorithms for other algorithmic problems, where geometric and topological ingredients can prove relevant. Here, a great pool of problems comes from problems for which efficient algorithms are known for planar graphs. While this line of work has been very active in the past decades, leading to extensive algorithmic techniques for graphs on surfaces for problems involving shortest paths [24], flows [29] or cuts [28], there are still many remaining open problems. For example, there is an $n^{O(g)}$ algorithm to compute an optimal SPARSEST CUT for graphs embedded on orientable surfaces of genus g [151] (generalizing the planar algorithm of Park and Philips [150]), but it is open whether the problem is fixed parameter tractable with respect to the genus. The MAX CUT problem is famously solvable in polynomial time for planar graphs [87], and this has been generalized for unweighted surface-embedded graphs in [73], but a general, combinatorial algorithm is still elusive. Computing the BRANCHWIDTH of a planar graph is solvable in polynomial time [170], but generalizing the algorithm to any other surface (even the projective plane or the torus) is still open (see Inkmann [99] for an interesting discussion of the challenges involved). In Chapter 4, we also developed techniques towards proving lower bounds for problems for graphs embedded on surfaces, and I am hopeful that our 4-Regular Graph Tiling problem might be used as a general reduction tool for more problems involving non-planar graphs.

Besides these classical algorithmic problems, it is also striking that many algorithmic questions specific to decompositions of graphs embedded on surfaces remain very unexplored. While our work [D] pretty much settled the exact parameterized complexity of the SHORTEST CUT GRAPH problem, it is still a tantalizing question whether we can improve on the $O(\log^2 g)-$approximation algorithm of Erickson and Har-Peled [63]. Other types of decompositions yield equally open questions. For example, a *pants decomposition* is a family of closed curves cutting a surface into a family of *pairs of pants*, i.e., spheres with

three boundaries. It is unknown whether computing the shortest pants decomposition of a surface is doable in polynomial time or whether the problem is **NP**-hard. Strikingly, this even seems to be the case in the case of the Euclidean plane with boundaries (see Eppstein [62]). Similarly, while, as we mentioned in Chapter 7, one can compute efficiently a canonical system of loops for a surface-embedded graph with good bounds on its length, the complexity of the problem of computing the shortest one is still unknown (see Erickson and Whittlesey [64]). A more general question is to investigate the best algorithms for the very general SHORTEST MAP EMBEDDING problem presented in Chapter 4: while it is W[1]-hard, it seems plausible that that a fixed parameter tractable approximation scheme similar to the one we presented for MULTICUT could exist.

Homotopy Height and sweep-outs. The main question that motivated most of the work presented in Chapter 5 is still open: can we compute the HOMOTOPY HEIGHT of a planar graph (i.e., any of the presented variants) in polynomial time? An even easier question also begs to be answered: in Section 5.4, we saw that a variant of the problem is fixed-parameter tractable when parameterized by the output, but the algorithm is unknown. Figuring out such an algorithm, even for the small values of the output, is a natural first step towards progress. A more disguised perspective is the following one: homotopy height can be perceived as an analogue of path-like width parameters (such as path-width or cut-width) with more topological structure: at any step of the decomposition, both sides of the decomposition form topological disks. In the case of tree-like width parameters, such a strong topological structure is already present for free for branch decompositions, as was demonstrated by Seymour and Thomas [170], and this was efficiently leveraged in order to design improved algorithms, see for example the sphere-cut decompositions introduced in [57]. Could it be that the topological structure of homotopy height makes it the correct parameter to design parameterized algorithms for specific problems on planar graphs?

As discussed in Chapter 5, one reason to be optimistic that HOMOTOPY HEIGHT may be solvable in polynomial time is that BRANCHWIDTH is, and they display a similar topological structure. The polynomial-time algorithm for BRANCHWIDTH on planar graphs leverages deep results in Robertson-Seymour theory [161], and developing an analogue of those for linear sweep-outs is a natural direction. Furthermore, our study of HOMOTOPY HEIGHT was partly motivated by similar Riemannian questions on the existence and the length of geodesics on spheres, and it turns out that tree-like sweepouts are also relevant for such questions, see for example Balacheff [8], Guth [84] and Liokumovitch [129]. In ongoing work with Alfredo Hubard and Francis Lazarus, we leverage the strong structural theory underlying branchwidth on planar graphs to prove new results on the geometry of Riemannian 2-spheres and obtain improved bounds on the lengths of their geodesics.

Algorithms and hardness in 3 dimensions. As the attentive reader probably noticed, none of the key questions mentioned in the introduction of Chapter 6 was actually solved in that chapter, and for the fundamental problems of UNKNOT RECOGNITION, KNOT EQUIVALENCE or UNKNOTTING NUMBER, the gaps between the computational lower bounds and upper bounds are still significant and beg to be investigated further. Let us insist again

that for UNKNOTTING NUMBER, no lower bound is known but the problem is not even known to be decidable: from that perspective, a key natural step for our hardness reductions is to strengthen them so that they apply to *knots* instead of *links*, thus bridging the gap between UNLINKING NUMBER and UNKNOTTING NUMBER. While we succeeded to do so for the problem of finding the optimal number of Reidemeister moves to untangle an unknot, new techniques seem to be required as our arguments currently rely heavily on linking numbers. On the side of upper bounds, at the time of writing, it is expected that the new quasi-polynomial time algorithm for UNKNOT RECOGNITION announced by Lackenby [123] will pave the way towards a host of new results in algorithmic knot theory. Yet, as a polynomial-time algorithm still seems out of reach, a natural line of inquiry lies here again on parameterized algorithms: given a knot diagram of bounded treewidth, can we detect in polynomial time whether it is an unknot? This question seems to be still open (see [17] for an algorithm for knot diagrams of treewidth 2), despite the fact that many knot invariants are fixed parameter tractable when parameterized by the treewidth, such as the Jones polynomial [130] (which is conjectured to detect the unknot), the HOMFLY-PT polynomial [22] or some quantum knot invariants [131]. A fascinating open question is whether stronger invariants of homological nature, such as Khovanov Homology [106] or Knot Floer Homology [148, 156] can also be computed in polynomial time for knot diagrams of bounded treewidth– this would be particularly interesting since those are known to detect the unknot [113] (see Bar-Natan [9] for a proposed approach).

Finally, it is also worthwhile to work the other way around and try to identify simple 3-dimensional problems for which we *can* design (somewhat) efficient algorithms. Colin de Verdière and Parsa [49, 50] launched the investigation of the complexity of deciding whether a (possibly non-simple) closed curve on the boundary of a 3-manifold is contractible, providing an exponential-time algorithm on it. In very recent work with Chambers, Lazarus and Parsa [O], we improved on their algorithm, showing that the problem is in **NP**, and more strikingly showing that the problem is fixed parameter tractable when parameterized by the underlying 3-manifold: in other terms, after a fixed exponential-time preprocessing depending only on the 3-manifold M one can solve any number of contractibility tests for a curve on the boundary of the manifold in polynomial time. The next steps are to devise a more general free homotopy test under the same conditions and running times, and to tackle the case of a curve not on the boundary of the manifold (although we expect that more general problem to be *much* more difficult).

Bibliography

The publications that I have co-authored are listed at the beginning of this document, with alphabetic keys.

[1] S. I. ADYAN, *Unsolvability of some algorithmic problems in the theory of groups*, Trudy Moskov. Mat. Obšc, 6 (1957), pp. 231–298. Cited on page 57

[2] I. AGOL, J. HASS, AND W. THURSTON, *The computational complexity of knot genus and spanning area*, Transactions of the American Mathematical Society, 358 (2006), pp. 3821–3850. Cited on page 62

[3] S. B. AKERS JR, *The use of wye-delta transformations in network simplification*, Operations Research, 8 (1960), pp. 311–323. Cited on page 80

[4] H. ALT AND M. GODAU, *Computing the Fréchet distance between two polygonal curves*, International Journal of Computational Geometry & Applications, 5 (1995), pp. 75–91. Cited on page 44

[5] D. ARCHDEACON AND C. P. BONNINGTON, *Two maps on one surface*, Journal of Graph Theory, 36 (2001), pp. 198–216. Cited on page 77

[6] S. ARORA, *Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems*, Journal of the ACM (JACM), 45 (1998), pp. 753–782. Cited on page 28

[7] S. ARORA AND B. BARAK, *Computational complexity: a modern approach*, Cambridge University Press, 2009. Cited on page 13, 19

[8] F. BALACHEFF, *Measurements of Riemannian two-disks and two-spheres*, Pacific Journal of Mathematics, 275 (2015), pp. 167–181. Cited on page 92

[9] D. BAR-NATAN, *Fast Khovanov homology computations*, Journal of Knot Theory and Its Ramifications, 16 (2007), pp. 243–255. Cited on page 93

[10] L. BARRIÈRE, P. FLOCCHINI, F. V. FOMIN, P. FRAIGNIAUD, N. NISSE, N. SANTORO, AND D. M. THILIKOS, *Connected graph searching*, Information and Computation, 219 (2012), pp. 1–16. Cited on page 45

[11] E. A. BENDER AND E. R. CANFIELD, *The asymptotic number of rooted maps on a surface*, Journal of Combinatorial Theory, Series A, 43 (1986), pp. 244–257. Cited on page 79

[12] S. BESPAMYATNIKH, *Computing homotopic shortest paths in the plane*, Journal of Algorithms, 49 (2003), pp. 284–303. Cited on page 79

[13] D. BIENSTOCK, *Graph searching, path-width, tree-width and related problems (a survey)*, in Reliability Of Computer And Communication Networks, 1989. Cited on page 45

[14] R. H. BING, *The geometric topology of 3-manifolds*, AMS, Providence, RI, 1983. Cited on page 69

[15] G. D. BIRKHOFF, *Dynamical systems with two degrees of freedom*, Proceedings of the National Academy of Sciences of the United States of America, 3 (1917), p. 314. Cited on page 46

[16] S. A. BLEILER, *A note on unknotting number*, in Mathematical Proceedings of the Cambridge Philosophical Society, vol. 96, Cambridge University Press, 1984, pp. 469–471. Cited on page 59

[17] H. L. BODLAENDER, B. BURTON, F. V. FOMIN, AND A. GRIGORIEV, *Knot diagrams of treewidth two*, in International Workshop on Graph-Theoretic Concepts in Computer Science, Springer, 2020, pp. 80–91. Cited on page 84, 93

[18] W. W. BOONE, *The word problem*, Proceedings of the National Academy of Sciences of the United States of America, 44 (1958), p. 1061. Cited on page 57

[19] G. BORRADAILE, P. KLEIN, AND C. MATHIEU, *An $O(n \log n)$ approximation scheme for Steiner tree in planar graphs*, ACM Transactions on Algorithms (TALG), 5 (2009), pp. 1–31. Cited on page 29

[20] G. R. BRIGHTWELL AND P. WINKLER, *Submodular percolation*, SIAM Journal on Discrete Mathematics, 23 (2009), pp. 1149–1178. Cited on page 45

[21] B. BURTON, H. EDELSBRUNNER, J. ERICKSON, AND S. TILLMANN, eds., *Computational geometric and algebraic topology. Abstracts from the workshop held October 11–17, 2015.*, vol. 12, European Mathematical Society (EMS) Publishing House, Zurich; Mathematisches Forschungsinstitut Oberwolfach, Oberwolfach, 2015. Cited on page 84

[22] B. A. BURTON, *The HOMFLY-PT polynomial is Fixed-Parameter Tractable*, in 34th International Symposium on Computational Geometry (SoCG 2018), Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018. Cited on page 84, 93

[23] ——, *The next 350 million knots*, in 36th International Symposium on Computational Geometry (SoCG 2020), Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020. Cited on page 85

[24] S. CABELLO, E. W. CHAMBERS, AND J. ERICKSON, *Multiple-source shortest paths in embedded graphs*, SIAM Journal on Computing, 42 (2013), pp. 1542–1571. Cited on page 32, 91

[25] S. CABELLO, Y. LIU, A. MANTLER, AND J. SNOEYINK, *Testing homotopy for paths in the plane*, Discrete & Computational Geometry, 31 (2004), pp. 61–81. Cited on page 79

[26] E. W. CHAMBERS, É. COLIN DE VERDIÈRE, J. ERICKSON, S. LAZARD, F. LAZARUS, AND S. THITE, *Homotopic Fréchet distance between curves — or, walking your dog in the woods in polynomial time*, Computational Geometry: Theory and Applications, 43 (2010), pp. 295–311. Cited on page 44

[27] E. W. CHAMBERS, É. COLIN DE VERDIÈRE, J. ERICKSON, F. LAZARUS, AND K. WHITTLESEY, *Splitting (complicated) surfaces is hard*, Computational Geometry: Theory and Applications, 41 (2008), pp. 94–110. Cited on page 32

[28] E. W. CHAMBERS, J. ERICKSON, K. FOX, AND A. NAYYERI, *Minimum cuts in surface graphs*, arXiv preprint arXiv:1910.04278, (2019). Cited on page 91

[29] E. W. CHAMBERS, J. ERICKSON, AND A. NAYYERI, *Homology flows, cohomology cuts*, SIAM Journal on Computing, 41 (2012), pp. 1605–1634. Cited on page 91

[30] E. W. CHAMBERS AND D. LETSCHER, *On the height of a homotopy*, in Proceedings of the 21st Canadian Conference on Computational Geometry (CCCG), 2009, pp. 103–106. Cited on page 43, 45, 48

[31] G. R. CHAMBERS AND Y. LIOKUMOVICH, *Converting homotopies to isotopies and dividing homotopies in half in an effective way*, Geometric and Functional Analysis, 24 (2014), pp. 1080–1100. Cited on page 47, 48

[32] G. R. CHAMBERS AND R. ROTMAN, *Contracting loops on a Riemannian 2-surface*. arXiv 1311.2995v2, 2013. Cited on page 48, 49

[33] H. CHANG, M. COSSARINI, AND J. ERICKSON, *Lower bounds for electrical reduction on surfaces*, in 35th International Symposium on Computational Geometry,

SoCG 2019, June 18-21, 2019, Portland, Oregon, USA, G. Barequet and Y. Wang, eds., vol. 129 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019, pp. 25:1–25:16. Cited on page 82

[34] H.-C. CHANG, *Tightening curves and graphs on surfaces*, PhD thesis, University of Illinois at Urbana-Champaign, 2018. Cited on page 80

[35] H.-C. CHANG, J. ERICKSON, AND C. XU, *Detecting weakly simple polygons*, in Proceedings of the twenty-sixth annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2014, pp. 1655–1670. Cited on page 80

[36] É. COLIN DE VERDIÈRE, *Topological algorithms for graphs on surfaces*, PhD thesis, École normale supérieure, 2012. Habilitation thesis, available at `http://www.di.ens.fr/~colin/`. Cited on page 32

[37] É. COLIN DE VERDIÈRE, *Multicuts in planar and bounded-genus graphs with bounded number of terminals*, Algorithmica, 78 (2017), pp. 1206–1224. Preliminary version in *Proceedings of the 23rd European Symposium on Algorithms (ESA)*, 2015, and in arXiv:1502.00911. Cited on page 24, 27, 32, 33

[38] É. COLIN DE VERDIÈRE AND J. ERICKSON, *Tightening nonsimple paths and cycles on surfaces*, SIAM Journal on Computing, 39 (2010), pp. 3784–3813. Cited on page 16, 78, 79, 90

[39] Y. COLIN DE VERDIÈRE, *Comment rendre géodésique une triangulation d'une surface ?*, L'Enseignement Mathématique, 37 (1991), pp. 201–212. Cited on page 75, 78

[40] P. F. CORTESE, G. DI BATTISTA, M. PATRIGNANI, AND M. PIZZONIA, *On embedding a cycle in a plane graph*, Discrete Mathematics, 309 (2009), pp. 1856–1869. Cited on page 80

[41] M. COSSARINI, *Discrete surfaces with length and area and minimal fillings of the circle*, arXiv preprint arXiv:2009.02415, (2020). Cited on page 17

[42] A. COWARD AND M. LACKENBY, *An upper bound on Reidemeister moves*, American Journal of Mathematics, 136 (2014), pp. 1023–1066. Cited on page 58

[43] C. CROKE AND M. KATZ, *Universal volume bounds in Riemannian manifolds*, Surveys in Differential Geometry, 8 (2003), pp. 109–137. Cited on page 46

[44] M. CULLER, C. GORDON, J. LUECKE, AND P. B. SHALEN, *Dehn surgery on knots*, Annals of Mathematics, (1987), pp. 237–300. Cited on page 73

[45] M. CYGAN, F. V. FOMIN, L. KOWALIK, D. LOKSHTANOV, D. MARX, M. PILIPCZUK, M. PILIPCZUK, AND S. SAURABH, *Parameterized algorithms*, Springer, 2015. Cited on page 13, 19, 20, 35, 83

[46] E. DAHLHAUS, D. S. JOHNSON, C. H. PAPADIMITRIOU, P. D. SEYMOUR, AND M. YANNAKAKIS, *The complexity of multiterminal cuts*, SIAM Journal on Computing, 23 (1994), pp. 864–894. Cited on page 24, 26

[47] H. DE FRAYSSEIX, J. PACH, AND R. POLLACK, *Small sets supporting Fáry embeddings of planar graphs*, in Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC), 1988, pp. 426–433. Cited on page 55

[48] M. DE GRAAF AND A. SCHRIJVER, *Making curves minimally crossing by Reidemeister moves*, Journal of Combinatorial Theory, Series B, 70 (1997), pp. 134–156. Cited on page 79, 80

[49] É. C. DE VERDIÈRE AND S. PARSA, *Deciding contractibility of a non-simple curve on the boundary of a 3-manifold*, in Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2017, pp. 2691–2704. Cited on page 93

[50] ——, *Deciding contractibility of a non-simple curve on the boundary of a 3-manifold: A computational loop theorem*, arXiv preprint arXiv:2001.04747, (2020). Cited on page 93

[51] M. DEHN, *Transformation der Kurven auf zweiseitigen Flächen*, Mathematische Annalen, 72 (1912), pp. 413–421. Cited on page 44

[52] E. D. DEMAINE, A. C. HESTERBERG, AND J. S. KU, *Finding closed quasigeodesics on convex polyhedra*, arXiv preprint arXiv:2008.00589, (2020). Cited on page 90

[53] L. DEMASI AND B. MOHAR, *Four terminal planar delta-wye reducibility via rooted$K_{2,4}$ minors*, in Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2014, pp. 1728–1742. Cited on page 82

[54] M. DESBRUN, M. MEYER, AND P. ALLIEZ, *Intrinsic parameterizations of surface meshes*, in Eurographics conference proceedings, 2002, pp. 209–218. Cited on page 24

[55] V. DESPRÉ AND F. LAZARUS, *Computing the geometric intersection number of curves*, Journal of the ACM (JACM), 66 (2019), pp. 1–49. Cited on page 79

[56] D. DOLEV, T. LEIGHTON, AND H. TRICKEY, *Planar embedding of planar graphs*, Advances in Computing Research, 2 (1984), pp. 147–161. Cited on page 55

[57] F. DORN, E. PENNINKX, H. L. BODLAENDER, AND F. V. FOMIN, *Efficient exact algorithms on planar graphs: exploiting sphere cut branch decompositions*, Algorithmica, 58 (2010), pp. 790–810. Cited on page 92

[58] R. G. DOWNEY AND M. R. FELLOWS, *Parameterized complexity*, Springer Science & Business Media, 2012. Cited on page 20

[59] S. E. DREYFUS AND R. A. WAGNER, *The Steiner problem in graphs*, Networks, 1 (1972), pp. 195–207. Cited on page 25, 28, 30

[60] V. DUJMOVIĆ, M. R. FELLOWS, M. KITCHING, G. LIOTTA, C. MCCARTIN, N. NISHIMURA, P. RAGDE, F. ROSAMOND, S. WHITESIDES, AND D. R. WOOD, *On the parameterized complexity of layered graph drawing*, Algorithmica, 52 (2008), pp. 267–292. Cited on page 55

[61] J. A. ELLIS-MONAGHAN AND I. MOFFATT, *Graphs on surfaces: dualities, polynomials, and knots*, vol. 84, Springer, 2013. Cited on page 16

[62] D. EPPSTEIN, *Squarepants in a tree: sum of subtree clustering and hyperbolic pants decomposition*, ACM Transactions on Algorithms, 5 (2009). Cited on page 92

[63] J. ERICKSON AND S. HAR-PELED, *Optimally cutting a surface into a disk*, Discrete & Computational Geometry, 31 (2004), pp. 37–59. Cited on page 3, 9, 24, 29, 32, 33, 37, 91

[64] J. ERICKSON AND K. WHITTLESEY, *Greedy optimal homotopy and homology generators*, in Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2005, pp. 1038–1046. Cited on page 32, 92

[65] B. FARB AND D. MARGALIT, *A primer on mapping class groups*, Princeton University Press, 2011. Cited on page 78

[66] I. FÁRY, *On straight line representations of planar graphs*, Acta scientiarum mathematicarum (Szeged), 11 (1948), pp. 229–233. Cited on page 55, 75

[67] S. FELSMER, G. LIOTTA, AND S. WISMATH, *Straight-line drawings on restricted integer*, Graph Algorithms and Applications 4, 4 (2006), p. 363. Cited on page 55

[68] M. FILAKOVSKÝ, U. WAGNER, AND S. ZHECHEV, *Embeddability of simplicial complexes is undecidable*, in Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2020, pp. 767–785. Cited on page 61

[69] R. H. FOX, *On the imbedding of polyhedra in 3-space*, Annals of Mathematics, (1948), pp. 462–470. Cited on page 70

[70] M. H. FREEDMAN ET AL., *The topology of four-dimensional manifolds*, Journal of Differential Geometry, 17 (1982), pp. 357–453. Cited on page 2, 8

[71] R. FULEK AND C. D. TÓTH, *Crossing minimization in perturbed drawings*, Journal of Combinatorial Optimization, 40 (2020), pp. 279–302. Cited on page 80

[72] D. GABAI, *Foliations and the topology of 3-manifolds. iii*, Journal of Differential Geometry, 26 (1987), pp. 479–536. Cited on page 72

[73] A. GALLUCCIO, M. LOEBL, AND J. VONDRÁK, *Optimization via enumeration: a new algorithm for the max cut problem*, Mathematical Programming, 90 (2001), pp. 273–290. Cited on page 91

[74] M. GAREY AND D. S. JOHNSON, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, 1979. Cited on page 62

[75] M. R. GAREY AND D. S. JOHNSON, *Crossing number is NP-complete*, SIAM Journal on Algebraic Discrete Methods, 4 (1983), pp. 312–316. Cited on page 85

[76] N. GARG, V. V. VAZIRANI, AND M. YANNAKAKIS, *Primal-dual approximation algorithms for integral flow and multicut in trees*, Algorithmica, 18 (1997), pp. 3–20. Cited on page 26

[77] J. GEELEN, T. HUYNH, AND R. B. RICHTER, *Explicit bounds for graph minors*, Journal of Combinatorial Theory, Series B, 132 (2018), pp. 80–106. Cited on page 77

[78] J. R. GILBERT, J. P. HUTCHINSON, AND R. E. TARJAN, *A separator theorem for graphs of bounded genus*, Journal of Algorithms, 5 (1984), pp. 391–407. Cited on page 40

[79] C. GORDON AND J. LUECKE, *Knots are determined by their complements*, J. Amer. Math. Soc., 2 (1989), pp. 371–415. Cited on page 19, 58, 72

[80] C. M. GORDON AND R. LITHERLAND, *Incompressible planar surfaces in 3-manifolds*, Topology and its Applications, 18 (1984), pp. 121–144. Cited on page 72

[81] M. GROMOV, *Quantitative homotopy theory*, Prospects in Mathematics: Invited Talks on the Occasion of the 250th Anniversary of Princeton University (H. Rossi, ed.), (1999), pp. 45–49. Cited on page 46

[82] J. L. GROSS AND T. W. TUCKER, *Topological graph theory*, Wiley, 1987. Cited on page 16

[83] X. GU AND S.-T. YAU, *Global conformal surface parameterization*, in Proceedings of the Eurographics/ACM Symposium on Geometry Processing, 2003, pp. 127–137. Cited on page 24

[84] L. GUTH, *Lipschitz maps from surfaces*, Geometric and Functional Analysis, 15 (2005), pp. 1052–1090. Cited on page 92

[85] ——, *Recent progress in quantitative topology*, Surveys in Differential Geometry, 22 (2017), pp. 191–216. Cited on page 46

[86] L. GUTH, H. PARLIER, AND R. YOUNG, *Pants decompositions of random surfaces*, Geometric and Functional Analysis, 21 (2011), pp. 1069–1090. Cited on page 78

[87] F. HADLOCK, *Finding a maximum cut of a planar graph in polynomial time*, SIAM Journal on Computing, 4 (1975), pp. 221–225. Cited on page 91

[88] W. HAKEN, *Theorie der Normalflachen, ein Isotopiekriterium für den Kreisnoten*, Acta Mathematica, 105 (1961), pp. 245–375. Cited on page 58

[89] S. HAR-PELED, A. NAYYERI, M. SALAVATIPOUR, AND A. SIDIROPOULOS, *How to walk your dog in the mountains with no magic leash*, in Proceedings of the 28th Annual Symposium on Computational Geometry (SOCG), ACM, 2012, pp. 121–130. Cited on page 44, 51, 52

[90] ——, *How to walk your dog in the mountains with no magic leash*, Discrete & Computational Geometry, 55 (2016), pp. 39–73. Cited on page 45

[91] J. HASS, J. C. LAGARIAS, AND N. PIPPENGER, *The computational complexity of knot and link problems*, Journal of the ACM, 46 (1999), pp. 185–211. Cited on page 58

[92] J. HASS AND F. MORGAN, *Geodesic nets on the 2-sphere*, Proceedings of the American Mathematical Society, 124 (1996), pp. 3843–3850. Cited on page 90

[93] J. HASS AND P. SCOTT, *Intersections of curves on surfaces*, Israel Journal of Mathematics, 51 (1985), pp. 90–120. Cited on page 80

[94] ——, *Shortening curves on surfaces*, Topology, 33 (1994), pp. 25–43. Cited on page 79, 80

[95] A. HATCHER, *Algebraic topology*, Cambridge University Press, 2002. Available at `http://www.math.cornell.edu/~hatcher/`. Cited on page 13

[96] C. HAYASHI AND K. SHIMOKAWA, *Thin position of a pair (3-manifold, 1-submanifold)*, Pacific Journal of Mathematics, 197 (2001), pp. 301–324. Cited on page 85, 90

[97] J. HERSHBERGER AND J. SNOEYINK, *Computing minimum length paths of a given homotopy class*, Computational Geometry: Theory and Applications, 4 (1994), pp. 63–98. Cited on page 79

[98] J. HOPCROFT AND R. TARJAN, *Efficient planarity testing*, Journal of the ACM, 21 (1974), pp. 549–568. Cited on page 61

[99] T. INKMANN, *Tree-based decompositions of graphs on surfaces and applications to the Traveling Salesman Problem*, Georgia Institute of Technology, 2008. Cited on page 91

[100] A. ISHII, K. KISHIMOTO, H. MORIUCHI, AND M. SUZUKI, *A table of genus two handlebody-knots up to six crossings*, Journal of Knot Theory and Its Ramifications, 21 (2012), p. 1250035. Cited on page 72

[101] G. F. ITALIANO, Y. NUSSBAUM, P. SANKOWSKI, AND CH. WULFF-NILSEN, *Improved algorithms for Min Cut and Max Flow in undirected planar graphs*, in Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC), 2011, pp. 313–322. Cited on page 27

[102] S. V. IVANOV ET AL., *Recognizing the 3-sphere*, Illinois Journal of Mathematics, 45 (2001), pp. 1073–1117. Cited on page 61

[103] R. KARASEV AND A. SKOPENKOV, *Some converses' to intrinsic linking theorems*, arXiv preprint arXiv:2008.02523, (2020). Cited on page 61

[104] L. H. KAUFFMAN, *Formal knot theory*, Courier Corporation, 2006. Cited on page 85

[105] B. KERÉKJÁRTÓ, *Vorlesung über Topologie*, Springer-Verlag, 1923. Cited on page 19

[106] M. KHOVANOV, *A categorification of the Jones polynomial*, Duke Mathematical Journal, 101 (2000), pp. 359–426. Cited on page 93

[107] R. KIRBY, *A calculus for framed links in* $\mathbb{S}^3$, Inventiones mathematicae, 45 (1978), pp. 35–56. Cited on page 71

[108] P. N. KLEIN, *A subset spanner for planar graphs, with application to subset TSP*, in Proceedings of the thirty-eighth annual ACM symposium on Theory of computing, 2006, pp. 749–756. Cited on page 29

[109] ——, *A linear-time approximation scheme for tsp in undirected planar graphs with edge-weights*, SIAM Journal on Computing, 37 (2008), pp. 1926–1952. Cited on page 29

[110] P. N. KLEIN AND D. MARX, *Solving* PLANAR k-TERMINAL CUT *in* $O(n^{c\sqrt{k}})$ *time*, in Proceedings of the 39th International Colloquium on Automata, Languages and Programming (ICALP), 2012, pp. 569–580. Cited on page 27

[111] D. KOENIG AND A. TSVIETKOVA, *NP-hard problems naturally arising in knot theory*, arXiv preprint arXiv:1809.10334, (2018). Cited on page 60

[112] ——, *Unlinking, splitting, and some other NP-hard problems in knot theory*, in Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, 2021, pp. 1496–1507. Cited on page 60

[113] P. B. KRONHEIMER AND T. S. MROWKA, *Khovanov homology is an unknot-detector*, Publications mathématiques de l'IHÉS, 113 (2011), pp. 97–208. Cited on page 93

[114] G. KUPERBERG, *Knottedness is in NP, modulo GRH*, Advances in Mathematics, 256 (2014), pp. 493–506. Cited on page 58

[115] ——, *How hard is it to approximate the Jones polynomial?*, Theory of Computing, 11 (2015), pp. 183–219. Cited on page 60

[116] ——, *Algorithmic homeomorphism of 3-manifolds as a corollary of geometrization*, Pacific Journal of Mathematics, 301 (2019), pp. 189–241. Cited on page 58

[117] G. KUPERBERG AND E. SAMPERTON, *Coloring invariants of knots and links are often intractable*, Algebraic & Geometric Topology, (2021). To appear. Cited on page 60

[118] M. LACKENBY, *The crossing number of composite knots*, Journal of Topology, 2 (2009), pp. 747–768. Cited on page 85

[119] ——, *A polynomial upper bound on Reidemeister moves*, Annals of Mathematics, (2015), pp. 491–564. Cited on page 58

[120] ——, *The efficient certification of knottedness and Thurston norm*. To appear in Advances in Mathematics, 2016. Cited on page 58

[121] ——, *Some conditionally hard problems on links and 3-manifolds*, Discrete & Computational Geometry, 58 (2017), pp. 580–595. Cited on page 58, 59

[122] ——, *Links with splitting number one*, Geometriae Dedicata, (2021), pp. 1–33. Cited on page 59

[123] M. LACKENBY, *Unknot recognition in quasi-polynomial time*, 2021. Slides available on the author's webpage. Cited on page 58, 93

[124] F. LAZARUS, *Combinatorial graphs and surfaces from the computational and topological viewpoint followed by some notes on the isometric embedding of the square flat torus*, 2014. Habilitation Thesis, Université de Grenoble. Cited on page 17, 77

[125] F. LAZARUS, M. POCCHIOLA, G. VEGTER, AND A. VERROUST, *Computing a canonical polygonal schema of an orientable triangulated surface*, in Proceedings of the 17th Annual Symposium on Computational Geometry (SOCG), ACM, 2001, pp. 80–89. Cited on page 76

[126] F. LAZARUS AND J. RIVAUD, *On the homotopy test on surfaces*, in Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS), 2012, pp. 440–449. Cited on page 57

[127] A. S. LEVINE, *Slicing mixed Bing–Whitehead doubles*, Journal of Topology, 5 (2012), pp. 713–726. Cited on page 66

[128] B. LÉVY AND J.-L. MALLET, *Non-distorted texture mapping for sheared triangulated meshes*, in Proceedings of the 25th Annual Conference on Computer Graphics (SIGGRAPH), 1998, pp. 343–352. Cited on page 24

[129] Y. LIOKUMOVICH, *Slicing a 2-sphere*, Journal of Topology and Analysis, 06 (2014), pp. 573–590. Cited on page 92

[130] J. A. MAKOWSKY AND J. P. MARIÑO, *The parametrized complexity of knot polynomials*, Journal of Computer and System Sciences, 67 (2003), pp. 742–756. Cited on page 84, 93

[131] C. MARIA, *Parameterized complexity of quantum knot invariants*, in 37th International Symposium on Computational Geometry (SoCG 2021), Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021. Cited on page 84, 93

[132] A. A. MARKOV, *Unsolvability of certain problems in topology*, in Doklady Akademii Nauk, vol. 123, Russian Academy of Sciences, 1958, pp. 978–980. Cited on page 57

[133] F. C. MARQUES AND A. NEVES, *Min-max theory and the Willmore conjecture*, Annals of mathematics, (2014), pp. 683–782. Cited on page 90

[134] D. MARX, *Can you beat treewidth?*, in 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07), IEEE, 2007, pp. 169–179. Cited on page 38, 39

[135] ——, *A tight lower bound for planar multiway cut with fixed number of terminals*, in International Colloquium on Automata, Languages, and Programming, Springer, 2012, pp. 677–688. Cited on page 27, 32, 33, 34, 35

[136] J. MATOUŠEK, E. SEDGWICK, M. TANCER, AND U. WAGNER, *Untangling two systems of noncrossing curves*, in Graph Drawing, S. Wismath and A. Wolff, eds., vol. 8242 of Lecture Notes in Computer Science, Springer International Publishing, 2013, pp. 472–483. Cited on page 77

[137] J. MATOUŠEK, E. SEDGWICK, M. TANCER, AND U. WAGNER, *Embeddability in the 3-sphere is decidable*, Journal of the ACM (JACM), 65 (2018), pp. 1–49. Cited on page 61, 69

[138] J. MATOUSŠEK, M. TANCER, AND U. WAGNER, *Hardness of embedding simplicial complexes in R^d*, Journal of the European Mathematical Society, 13 (2011), pp. 259–295. Cited on page 61

[139] M. MIRZAKHANI, *Growth of Weil-Petersson volumes and random hyperbolic surface of large genus*, Journal of Differential Geometry, 94 (2013), pp. 267–300. Cited on page 78

[140] J. S. B. MITCHELL, *Guillotine subdivisions approximate polygonal subdivisions: a simple polynomial-time approximation scheme for geometric TSP, k-MST, and related problems*, SIAM Journal on Computing, 28 (1999), pp. 1298–1309. Cited on page 28

[141] B. MOHAR, *The genus crossing number*, ARS Mathematica Contemporanea, 2 (2009), pp. 157–162. Cited on page 77

[142] B. MOHAR AND C. THOMASSEN, *Graphs on surfaces*, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, 2001. Cited on page 13, 16

[143] E. E. MOISE, *Geometric topology in dimensions 2 and 3*, Springer-Verlag, 1977. Cited on page 19

[144] X. MUÑOZ, W. UNGER, AND I. VRT'O, *One sided crossing minimization is NP-hard for sparse graphs*, in International Symposium on Graph Drawing, Springer, 2001, pp. 115–123. Cited on page 86

[145] Y. NAKANISHI, *Unknotting numbers and knot diagrams with the minimum crossings*, Math. Sem. Notes Kobe Univ, 11 (1983), pp. 257–258. Cited on page 59

[146] S. NEGAMI, *Crossing numbers of graph embedding pairs on closed surfaces*, Journal of Graph Theory, 36 (2001), pp. 8–23. Cited on page 77

[147] P. S. NOVIKOV, *Über die algorithmische Unentscheidbarkeit des Wortproblems in der Gruppentheorie*. Tr. Mat. Inst. Steklova 44, 140 S. (1955)., 1955. Cited on page 57

[148] P. OZSVÁTH AND Z. SZABÓ, *Holomorphic disks and knot invariants*, Advances in Mathematics, 186 (2004), pp. 58–116. Cited on page 93

[149] J. PARDON, *On the distortion of knots on embedded surfaces*, Annals of mathematics, (2011), pp. 637–646. Cited on page 91

[150] J. K. PARK AND C. A. PHILLIPS, *Finding minimum-quotient cuts in planar graphs*, in Proceedings of the 25th Annual ACM Symposium on Theory of Computing (STOC), 1993, pp. 766–775. Cited on page 91

[151] V. PATEL, *Determining edge expansion and other connectivity measures of graphs of bounded genus*, in Proceedings of the 18th European Symposium on Algorithms (ESA), no. 6346 in Lecture Notes in Computer Science, 2010, pp. 561–572. Cited on page 91

[152] G. PERELMAN, *The entropy formula for the Ricci flow and its geometric applications*, arXiv preprint math/0211159, (2002). Cited on page 2, 8

[153] A. V. POGORELOV, *Quasi-geodesic lines on a convex surface*, Matematicheskii Sbornik, 67 (1949), pp. 275–306. Cited on page 90

[154] T. RADO, *Über den Begriff der Riemannschen Fläche*, Acta scientiarum mathematicarum (Szeged), 2 (1924), pp. 101–121. Cited on page 19

[155] A. A. RANICKI, *On the Hauptvermutung*, in The Hauptvermutung book, vol. 1 of K-Monogr. Math., Kluwer Acad. Publ., Dordrecht, 1996, pp. 3–31. Cited on page 19

[156] J. A. RASMUSSEN, *Floer homology and knot complements*, Harvard University, 2003. Cited on page 93

[157] K. REIDEMEISTER, *Elementare Begründung der Knotentheorie*, in Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg, vol. 5, Springer, 1927, pp. 24–32. Cited on page 18

[158] R. B. RICHTER AND G. SALAZAR, *Two maps with large representativity on one surface*, Journal of Graph Theory, 50 (2005), pp. 234–245. Cited on page 77

[159] N. ROBERTSON, P. SEYMOUR, AND R. THOMAS, *Quickly excluding a planar graph*, Journal of Combinatorial Theory, Series B, 62 (1994), pp. 323–348. Cited on page 52

[160] ——, *Sachs' linkless embedding conjecture*, Journal of Combinatorial Theory, Series B, 64 (1995), pp. 185–227. Cited on page 80

[161] N. ROBERTSON AND P. D. SEYMOUR, *Graph minors. XI. Circuits on a surface*, Journal of Combinatorial Theory, Series B, 60 (1994), pp. 72–106. Cited on page 92

[162] ——, *Graph minors. XIII. The disjoint paths problem*, Journal of Combinatorial Theory, Series B, 63 (1995), pp. 65–110. Cited on page 55

[163] ——, *Graph minors. XX. Wagner's conjecture*, Journal of Combinatorial Theory, Series B, 92 (2004), pp. 325–357. Cited on page 55

[164] D. ROLFSEN, *Knots and links*, Publish or Perish, 1976. Cited on page 13, 19, 70

[165] M. SCHAEFER, *Crossing Numbers of Graphs*, CRC Press, 2018. Cited on page 85

[166] M. SCHAEFER AND D. ŠTEFANKOVIČ, *The degenerate crossing number and higher-genus embeddings*, in International Symposium on Graph Drawing, Springer, 2015, pp. 63–74. Cited on page 90

[167] S. SCHLEIMER, *Sphere recognition lies in NP*, Low-dimensional and symplectic topology, 82 (2011), pp. 183–213. Cited on page 61, 70

[168] J. SCHULTENS, *Introduction to 3-manifolds*, vol. 151, American Mathematical Soc., 2014. Cited on page 13

[169] P. SEYMOUR AND R. THOMAS, *Graph searching and a min-max theorem for tree-width*, Journal of Combinatorial Theory, Series B, 58 (1993), pp. 22–33. Cited on page 45

[170] P. D. SEYMOUR AND R. THOMAS, *Call routing and the ratcatcher*, Combinatorica, 14 (1994), pp. 217–241. Cited on page 21, 45, 54, 84, 91, 92

[171] A. SKOPENKOV, *Extendability of simplicial maps is undecidable*, arXiv preprint arXiv:2008.00492, (2020). Cited on page 61

[172] S. SMALE, *Generalized Poincare's conjecture in dimensions greater than four*, Annals of Mathematics, (1961), pp. 391–406. Cited on page 2, 8

[173] A. SONG, *Existence of infinitely many minimal hypersurfaces in closed manifolds*, arXiv preprint arXiv:1806.08816, (2018). Cited on page 90

[174] E. STEINITZ, *Polyeder und Raumeinteilungen*, Encyclopädie der Mathematischen Wissenschaften, Band 3 (Geometrie) Teil 3AB12 (1922), pp. 1–139. Cited on page 80

[175] K. STEPHENSON, *Introduction to circle packing, the theory of discrete analytic functions*, Cambridge University Press, Moscow, 2005. Cited on page 76

[176] J. STILLWELL, *Classical topology and combinatorial group theory*, Springer-Verlag, New York, 1980. Cited on page 13

[177] R. TAMASSIA, ed., *Handbook of graph drawing and visualization*, CRC Press, 2013. Cited on page 75

[178] A. TURING, *Solvable and unsolvable problems*, Science News, 31 (1954), pp. 7–23. Cited on page 58

[179] W. T. TUTTE, *How to draw a graph*, Proceedings of the London Mathematical Society, 3 (1963), pp. 743–767. Cited on page 75

[180] V. V. VAZIRANI, *Approximation algorithms*, Springer Science & Business Media, 2013. Cited on page 19, 83

[181] L. VISMARA, *Planar straight-line drawing algorithms*, in Handbook of Graph Drawing and Visualization, CRC Press, 2013. Cited on page 55

[182] J. VYGEN, *Faster algorithm for optimum Steiner trees*, Information Processing Letters, 111 (2011), pp. 1075–1079. Cited on page 30

[183] V. V. WILLIAMS, *On some fine-grained questions in algorithms and complexity*, in Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018, World Scientific, 2018, pp. 3447–3487. Cited on page 20

[184] Z. WOOD, H. HOPPE, M. DESBRUN, AND P. SCHRÖDER, *Removing excess topology from isosurfaces*, ACM Transactions on Graphics, 23 (2004), pp. 190–208. Cited on page 24

[185] R. ZENTNER ET AL., *Integer homology* 3*-spheres admit irreducible representations in* $SL(2, \mathbb{C})$, Duke Mathematical Journal, 167 (2018), pp. 1643–1712. Cited on page 61