



**HAL**  
open science

# Topological embeddings of graphs on simplicial complexes

Thomas Magnard

► **To cite this version:**

Thomas Magnard. Topological embeddings of graphs on simplicial complexes. Data Structures and Algorithms [cs.DS]. Université Gustave Eiffel, 2021. English. NNT : 2021UEFL2022 . tel-03582537

**HAL Id: tel-03582537**

**<https://theses.hal.science/tel-03582537v1>**

Submitted on 21 Feb 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## Plongements topologiques de graphes sur les complexes simpliciaux

### **Thèse de doctorat de l'Université Gustave Eiffel**

École doctorale n° 532 Mathématiques et STIC, MSTIC

Spécialité de doctorat: Informatique

Unité de recherche : Laboratoire d'Informatique Gaspard-Monge  
UMR 8049

**Thèse présentée et soutenue à l'Université  
Gustave Eiffel, le 15/11/2021, par**

**Thomas MAGNARD**

#### **Composition du Jury**

**Dimitrios THILIKOS  
TOULOUPAS**

Directeur de recherche  
CNRS, LIRMM Montpellier

Rapporteur

**Dominique ATTALI**  
Directrice de recherche  
CNRS, GIPSA-lab, Grenoble

Examinatrice

**Cyril GAVOILLE**  
Professeur, Université de  
Bordeaux

Examineur

**Arnaud DE MESMAY**  
Chargé de recherche CNRS,  
LIGM, Marne-la-Vallée

Examineur

**Monique TEILLAUD**  
Directrice de recherche,  
INRIA Nancy-Grand Est,  
Loria

Présidente du jury

**Éric COLIN DE VERDIÈRE**  
directeur de recherche  
CNRS, LIGM, Marne-la-  
Vallée

Directeur de thèse

**Autre rapporteur**  
**Petr HLINENY** Professor,  
Masaryk University,  
République tchèque

Rapporteur

# Topological embeddings of graphs on simplicial complexes

## **Doctoral thesis of Université Gustave Eiffel**

Doctoral school n° 532: Mathématiques et STIC, MSTIC

Doctoral speciality: computer sciences

Research unit: Laboratoire d'Informatique Gaspard-Monge  
UMR 8049

**Thesis defended at Université Gustave Eiffel,  
on the 15/11/2021, by**

**Thomas MAGNARD**

### **Committee members**

**Dimitrios THILIKOS  
TOULOUPAS**

Directeur de recherche  
CNRS, LIRMM Montpellier

Referee

**Dominique ATTALI**  
Directrice de recherche  
CNRS, GIPSA-lab, Grenoble

Examiner

**Cyril GAVOILLE**  
Professeur, Université de  
Bordeaux

Examiner

**Arnaud DE MESMAY**  
Chargé de recherche CNRS,  
LIGM, Marne-la-Vallée

Examiner

**Monique TEILLAUD**  
Directrice de recherche,  
INRIA Nancy-Grand Est,  
Loria

Committee President

**Éric COLIN DE VERDIÈRE**  
directeur de recherche  
CNRS, LIGM, Marne-la-  
Vallée

Thesis supervisor

**Other reviewer**  
**Petr HLINENY** Professor,  
Masaryk University,  
République tchèque

Referee

**Keywords:** computational topology, embeddings, simplicial complexes, graphs, surfaces, fixed-parameter tractability

**Mots clés :** topologie algorithmique, graphes, surfaces, plongements, complexité paramétrée



# Résumé

L'étude des plongements topologiques de graphes, c'est-à-dire des manières de dessiner sans croisement un graphe dans un espace topologique, constitue un domaine classique à l'interface des mathématiques et de l'informatique dans les communautés de topologie, théorie topologique des graphes, topologie algorithmique et dessin de graphes.

Il est naturel de s'intéresser à la plongeabilité des graphes sur des espaces topologiques généraux tels que le plan, les surfaces ou des espaces de plus grande dimension. Or, comme tous les graphes sont plongeables dans un espace tridimensionnel, les espaces topologiques pour lesquels la question est non triviale sont de dimension au plus deux. Il est ainsi logique de considérer comme classe d'espaces la classe des complexes simpliciaux de dimension au plus deux, les espaces obtenus en recollant des sommets, des arêtes et des triangles. En particulier, ces espaces topologiques contiennent les surfaces, et le problème est déjà NP-difficile même en se restreignant aux surfaces.

Cette thèse présente deux algorithmes décidant le problème de la plongeabilité d'un graphe sur un 2-complexe. Les deux algorithmes fonctionnent en temps polynomial en la taille du graphe lorsque le complexe est fixé, mais seul le second est "fixed parameter tractable" quand paramétré par la taille du 2-complexe donné en entrée.

Le premier algorithme est basé sur des arguments d'ordre topologique. Sa stratégie consiste à réduire le problème de la plongeabilité d'un graphe sur un 2-complexe à un problème d'extension de plongement d'un graphe sur une surface pour lequel il existait déjà un algorithme dû à B.Mohar. De plus, dans le même temps, cette approche montre aussi que le problème est dans NP.

Le second algorithme est basé sur des arguments d'algorithmique des graphes. Il commence par retirer itérativement du graphe des sommets inutiles jusqu'à ce que le graphe ait une largeur de branche bornée. Ensuite, il utilise une stratégie de programmation dynamique qui décide si un graphe de largeur de branche bornée est plongeable sur un 2-complexe.



# Abstract

The study of topological embeddings of graphs, that is to say the ways of drawing without crossing a graph in a topological space, constitutes a classic field at the interface of mathematics and computer science in communities of topology, topological graph theory, computational topology and graph drawing.

It is natural to be interested in the embeddability of graphs on general topological spaces such as the plane, surfaces or larger spaces. However, as all graphs can be embedded into a three-dimensional space, the topological spaces for which the question is non-trivial are of dimension at most two. It is thus logical to consider the class of simplicial complexes of dimension at most two, the spaces obtained by gluing together vertices, edges and triangles. In particular, these topological spaces include the class of all surfaces and the problem is already NP-hard even if we restrict ourselves to surfaces.

This thesis presents two algorithms deciding the problem of the embeddability of a graph on a 2-complex. Both algorithms operate in polynomial time in the size of the graph when the complex is fixed, but only the second is fixed parameter tractable when parameterized by the size of the 2-complex given as input.

The first algorithm is based on topological arguments. Its strategy consists in reducing the problem of the embeddability of a graph on a 2-complex to an embedding extension problem of a graph on a surface for which there already existed an algorithm due to B.Mohar. Moreover, at the same time, this approach also shows that the problem is in NP.

The second algorithm is based on graph algorithmic arguments. It begins by iteratively removing unnecessary vertices from the graph until the graph has a bounded branch width. Then, it uses a dynamic programming strategy which decides whether a graph of bounded branch width is embeddable on a 2-complex.





# Contents

<b>Résumé</b>	iii
<b>Abstract</b>	v
<b>Contents</b>	vii
<b>1 Introduction en français</b>	<b>1</b>
1.1 Motivation	3
1.2 Domaines reliés	4
1.3 Contributions de la thèse	5
1.4 Organisation du reste de la thèse	7
<b>2 Introduction in English</b>	<b>9</b>
2.1 Motivation	11
2.2 Related domains	12
2.3 Contributions of this thesis	13
2.4 Organization of the rest of the thesis	14
<b>3 Preliminaries</b>	<b>17</b>
3.1 Topological spaces	17
3.2 Graphs	18
3.3 Simplicial complexes	22
3.4 Surfaces	24
3.5 Fixed parameter tractability	30
<b>4 Topological embedding problems</b>	<b>41</b>
4.1 Topological embeddings on simplicial complexes	41
4.2 Related problems	47
<b>5 Embeddings on 2-complexes: motivations and basic tools</b>	<b>51</b>
5.1 Motivations	51
5.2 Basic tools	56

<b>6</b>	<b>Embedding graphs on 2-complexes: a first algorithm</b>	<b>61</b>
6.1	Preliminaries	61
6.2	Reduction to EEPs on a pure 2-complex	63
6.3	Reduction to an EEP on a possibly disconnected surface	65
6.4	Reduction to a cellular EEP on a surface	70
6.5	Solving a cellular EEP on a surface	77
6.6	Proof of Theorems 6.1.1 and 6.1.2	77
<b>7</b>	<b>Embedding graphs on 2-complexes: a fixed-parameter tractable algorithm</b>	<b>79</b>
7.1	Overview and structure of the chapter	79
7.2	Preprocessing and data structures on 2-complexes	81
7.3	Partitioning graphs	87
7.4	Dynamic programming algorithm	95
7.5	Reduction to proper cellular embeddings	99
7.6	Algorithm for bounded branchwidth: Proof of Theorem 7.1.2	102
7.7	Reduction to bounded branchwidth: Proof of Theorem 7.1.3	103
<b>8</b>	<b>Conclusion</b>	<b>109</b>
	<b>Bibliography</b>	<b>113</b>

## Introduction en français

Il existe une énigme mathématique connue publiée pour la première fois par Henry Dudeney [Dud13], l'énigme des 3 maisons ou énigme de l'eau, l'électricité et le gaz: on a 3 maisons que l'on veut relier à une centrale électrique, un gazoduc et une source d'eau sans que les tuyaux ne se croisent (Figure 1.1). Ce problème est connu pour n'avoir aucune solution. Chercher à résoudre cette énigme revient à chercher à relier 3 points du plan (les maisons) à chacun des 3 autres points du plan (les centres de productions) par un chemin sans que ces chemins ne se croisent. En mathématiques, cet ensemble de 6 points dont 3 sont reliés aux 3 autres est connu sous le nom de  $K_{3,3}$ . De ce fait, l'énigme revient à chercher un dessin sans croisement de  $K_{3,3}$  sur le plan.

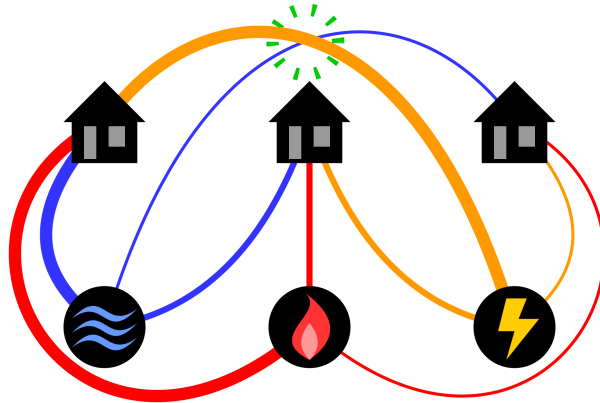


Figure 1.1: Un exemple de l'énigme des 3 maisons. Source: [Cmglee]

En mathématiques, un ensemble de points, appelés sommets, dont certaines paires, appelées arêtes, sont reliées est appelé un graphe et un dessin sans croisement d'un graphe est appelé un plongement. Ainsi, sous des apparences anodines, l'énigme des 3 maisons est en fait un cas iconique d'un problème mathématique classique, la planarité de graphe: étant donné un graphe  $G$ , décider si il existe un plongement de  $G$  sur le plan.

L'absence de solution de l'énigme s'exprime donc en disant que  $K_{3,3}$  n'est pas planaire. De plus, le problème de plongement de graphe peut être résolu en temps linéaire en la taille du graphe [HT74]. Si la stratégie utilisée par J.Hopcroft et R.Tarjan est la première fonctionnant en temps linéaire, les problèmes de planarité de graphe ont été vastement étudiés [Pat06] et il existe de nombreuses autres méthodes pour décider si un graphe est planaire, certaines étant plus anciennes [Kur30] et d'autres étant plus rapides [FOR06]. De plus il y a de nombreux raffinements très proches de ce problème qui ont aussi été étudiés. Par exemple, à la place de plonger un graphe sur le plan, on peut chercher à placer les sommets du graphe sur les croisements d'une grille la plus petite possible de manière à, pour tout quadruplet de sommets  $u, v, w, x$  tel que les deux paires de sommets  $\{u, v\}$  et  $\{w, x\}$  sont des arêtes du graphe, ne pas avoir de croisement entre le segment reliant  $u$  et  $v$  et le segment reliant  $w$  et  $x$  [FPP90].

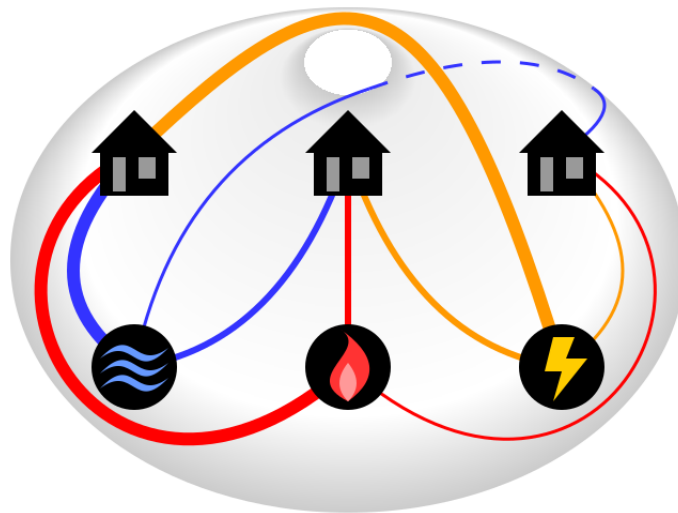


Figure 1.2: Une solution au problème des 3 maisons sur le tore. Source: [Cmglee]

De la même manière, la planarité de graphe est aussi un cas particulier d'un problème plus général, la plongeabilité d'un graphe sur une surface: étant donné un graphe  $G$  et une surface  $\mathcal{S}$ , décider si il existe un plongement de  $G$  sur  $\mathcal{S}$ . Les plongements que l'on considère sont des plongements topologiques, ce qui signifie que les arêtes peuvent être dessinées comme n'importe quelle courbe. Par exemple, bien que  $K_{3,3}$  ne soit pas dessinable sans croisement sur le plan, il est possible de le dessiner sans croisement sur la surface d'un objet avec un trou comme un tore (Figure 1.2). La plongeabilité d'un graphe sur une surface fixée peut être décidée en temps linéaire en la taille du graphe. Ainsi, B.Mohar [Moh99] et K.Kawarabayashi, B.Mohar et B.Reed [KMR08] tous deux présentent un algorithme qui décide la plongeabilité d'un graphe sur une surface travaillant en temps linéaire lorsque la surface est fixée. Par contre, dans les 2 cas la complexité de l'algorithme augmente très vite lorsque la complexité de la surface, ici son nombre de poignées, augmente. Cette explosion de la complexité de l'algorithme lorsque la surface se complexifie est attendue car la plongeabilité d'un graphe sur une surface

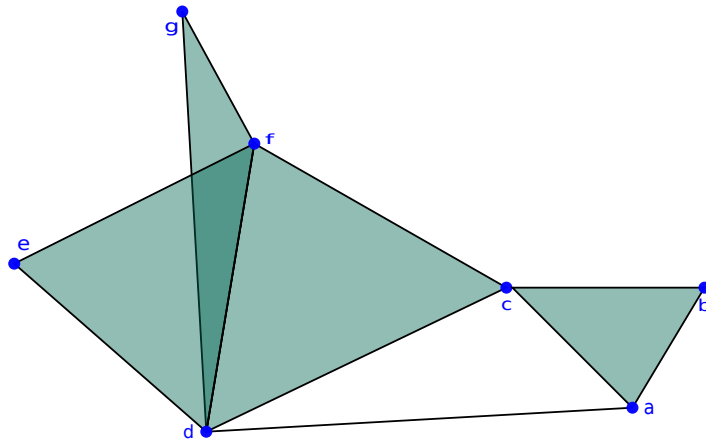


Figure 1.3: un 2-complexe dont les sommets sont  $\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}$  et  $\{g\}$ , les arêtes sont  $\{a, b\}, \{a, c\}, \{b, c\}, \{a, d\}, \{d, e\}, \{d, f\}, \{e, f\}, \{d, g\}, \{f, g\}, \{c, f\}$  et  $\{c, d\}$  et les triangles sont  $\{a, b, c\}, \{d, f, g\}, \{d, e, f\}$  et  $\{c, d, f\}$ .

arbitraire est **NP**-difficile [Tho89] et donc, il n'est pas raisonnable d'espérer trouver un algorithme dont le temps d'exécution est globalement polynomial (sinon cela prouverait que **P=NP**).

Lorsque l'on veut représenter une surface de manière finie, une méthode classique est de la représenter comme un ensemble de triangles collés les uns aux autres. Cependant, il est possible d'obtenir un ensemble de triangles collés les uns aux autres qui ne correspond pas à une surface. Par exemple, en collant trois triangles le long d'un même côté, on obtient une structure qui ne correspond à aucune surface. Du coup, on définit une structure nommée 2-complexe dont ces recollages sont des cas particuliers. Un 2-complexe est un ensemble de points dont certaines paires sont reliées par une arête et certains triplets sont les sommets d'un triangle tels que les sommets d'un triangle sont reliés deux à deux par des arêtes (Figure 1.3). Ainsi, la plongeabilité d'un graphe sur une surface est un cas particulier du problème étudié dans cette thèse, **la plongeabilité de graphe sur un 2-complexe: étant donné un graphe  $G$  et un 2-complexe  $\mathcal{C}$ , décider si il existe un plongement de  $G$  sur  $\mathcal{C}$** . La figure 1.4 montre un exemple de plongement d'un graphe sur un 2-complexe.

## 1.1 Motivation

Ce problème est très large. Ainsi, on pourrait encore élargir le problème par exemple aux 3-complexes en ajoutant dans  $\mathcal{C}$  des tétraèdres pleins entre 4 de ses points ou encore aux complexes de plus grande dimension en ajoutant des simplexes de dimension 4 ou plus mais alors tous les graphes peuvent se dessiner sur  $\mathcal{C}$  car tout graphe se plonge dans un tétraèdre plein et  $\mathcal{C}$  en contient au moins un. Donc, la plongeabilité d'un graphe sur un 2-complexe est le cas le plus large non trivial dans lequel on cherche à plonger un graphe sur une structure formée par des sommets, arêtes, triangles, tétraèdres plein

et leurs équivalent en plus grande dimension. De plus, il existe plusieurs problèmes très classiques de plongeabilité de graphes qui peuvent être réduits au problème de plongement de graphe sur un 2-complexe (cf Section 5.1.1) et en particulier la plongeabilité d'un graphe sur une surface.

De plus, les stratégies classiques utilisées pour construire un algorithme sur les surfaces ne s'appliquent pas aux 2-complexes (cf Section 5.1.2) et donc il n'y a pas de méthode évidente pour décider la plongeabilité d'un graphe sur un 2-complexe à partir des algorithmes pour le plongement de graphe sur les surfaces.

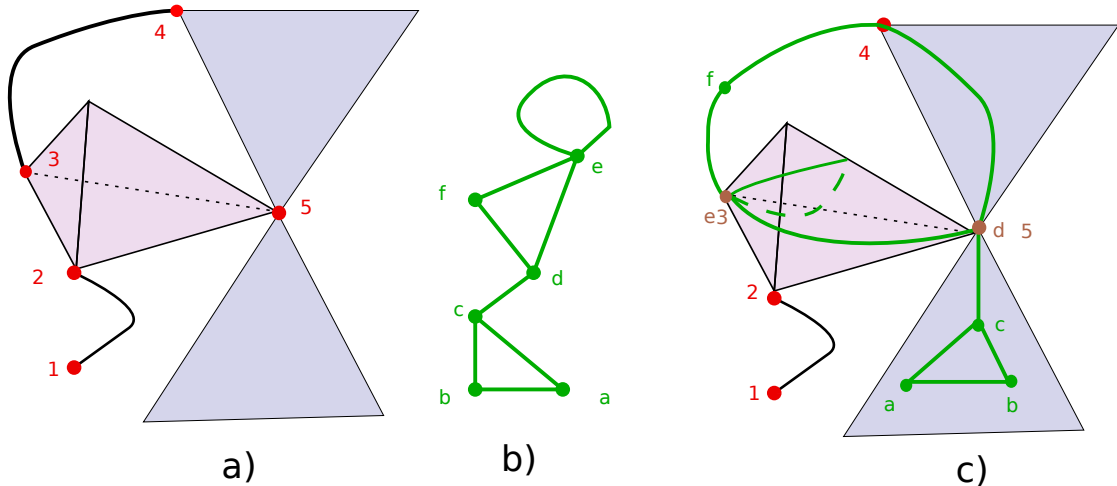


Figure 1.4: a): le 2-complexe  $\mathcal{C}$  formé par une sphère (un tétraèdre creux) et deux disques partageant un point, une arête isolée reliant la sphère à l'un des disques et une arête isolée attachée uniquement à la sphère. b): Un graphe  $G$  à plonger dessus c): Un plongement de  $G$  sur  $\mathcal{C}$ .

## 1.2 Domaines reliés

Les problèmes de plongements topologiques de graphes sont des problèmes clés de deux domaines mathématiques et informatiques: la théorie topologique des graphes et la topologie algorithmique.

**Théorie topologique des graphes** Le principe de la théorie topologique des graphes est d'étudier et d'utiliser les graphes vus comme des objets topologiques. Pour ce faire on utilise des représentations topologiques d'un graphe au travers d'un plongement ou d'un dessin de celui-ci, la plupart du temps sur une surface.

Ensuite, on peut utiliser les propriétés topologiques de ces représentations d'un graphe dans l'analyse de celui-ci. Ainsi, par exemple, on peut prouver que tous les graphes planaires sont 4-coloriables [AH76].

Inversement, on peut aussi utiliser les graphes afin d'obtenir une représentation combinatoire des espaces et des propriétés topologiques. Ensuite, grâce à cette représentation combinatoire, on peut étudier les espaces et les propriétés topologiques par le biais de la combinatoire. Par exemple, B.Mohar et C.Thomassen [MT01] présentent les éléments de base de l'analyse des surfaces et des plongements topologiques dessus comme des objets combinatoires utilisant des graphes plongés dessus.

Or, dans pour les deux cas, il est important de quantifier au mieux la complexité des graphes utilisés pour respectivement soit pouvoir en déduire le plus de propriétés possibles soit quantifier la complexité de l'espace représenté. Sachant que souvent les graphes sont étudiés en lien avec les surfaces, il est donc classique d'évaluer la complexité d'un graphe selon les surfaces sur lesquelles celui-ci est plongeable. Or c'est le nombre de poignées et de rubans de Möbius qui composent la surface (son genre) qui caractérise l'ensemble des graphes plongeables dessus. Il est donc très classique pour quantifier la complexité d'un graphe d'utiliser le genre minimal d'une surface sur lequel il est plongeable appelé le genre du graphe.

**Topologie algorithmique** [Veg04] Avec le développement de l'informatique, un domaine à l'interface de la topologie et de l'informatique est apparu: la topologie algorithmique, présentée ainsi pour l'une des premières fois par T.Dey, H.Edelsbrunner et S.Guha [DEG99]. L'idée est de chercher à construire des algorithmes qui calculent si des propriétés topologiques sont vérifiées ou construisent des objets ayant ces propriétés.

Les problèmes de plongements topologiques de graphe sont un cas particulier d'une classe clé de problème en topologie algorithmique: les problèmes de plongements topologiques généraux dont l'objet est de chercher à plonger un objet topologique sur un autre. Ainsi, il existe des problèmes centraux d'autres domaines de la topologie algorithmique qui peuvent être réduits à un cas de plongement topologique et donc utiliser des résultats connus pour les plongements afin d'obtenir un algorithme. Par exemple, le problème du nœud trivial qui est un problème central de la théorie des nœuds: étant donné un nœud dans l'espace, décider si il est un nœud trivial. Or, un nœud est trivial si et seulement si il existe un disque plongé dans  $\mathbb{R}^3$  dont le bord est le nœud [JT95]. Donc le problème du nœud trivial peut être réduit à un problème de plongement topologique.

De plus, les complexes sont une structure qui apparaît naturellement en topologie algorithmique. Par exemple, en topologie différentielle algorithmique, il est commun, étant donné un objet topologique  $\mathcal{T}$ , d'utiliser un complexe  $\mathcal{C}$  plongé bijectivement sur  $\mathcal{T}$ , appelé une triangulation de  $\mathcal{T}$ , afin d'étudier des problèmes sur  $\mathcal{T}$  ou de construire une triangulation de  $\mathcal{T}$  qui lui est difféomorphe [BM07]. Les triangulations sont ainsi des exemples de représentation d'un espace topologique par un complexe montrant comment ceux-ci, et en particulier les 2-complexes, peuvent apparaître naturellement.

### 1.3 Contributions de la thèse

Cette thèse décrit deux algorithmes qui, tous deux, prennent en entrée un graphe  $G$  et un 2-complexe  $\mathcal{C}$  et décident si il est possible de plonger  $G$  sur  $\mathcal{C}$ .



Le premier algorithme a été défini d'abord afin d'avoir un algorithme dont le temps d'exécution est polynomial quand le complexe est fixé. Cependant, dans cet algorithme, le degré du polynôme dépend de la taille du complexe; en terme technique, l'algorithme n'est pas FPT (*fixed parameter tractable*) paramétré par la taille du complexe. Or, avant cette thèse, on ne savait pas si il existe ou non un algorithme FPT en la taille du complexe décidant ce problème. Il était donc raisonnable de chercher un algorithme plus efficace. Ce qui est le cas du second algorithme qui lui est FPT en la taille du complexe.

Les 2 algorithmes sont totalement différents dans leurs stratégies pour résoudre le problème.

**Premier algorithme** Dans le premier, le principe est de construire une série de problèmes de plus en plus restreints où une partie du graphe est déjà plongé. Au départ, on a un graphe  $G$  que l'on veut plonger sur un complexe  $\mathcal{C}$ . On va plonger un sous-graphe de  $G$  de taille bornée sur  $\mathcal{C}$ . L'idée est d'utiliser des arguments d'ordre topologique pour restreindre de plus en plus la structure de  $\mathcal{C}$  et du plongement partiel dessus quitte à rajouter des arêtes et des sommets à  $G$  jusqu'à obtenir un problème de plongement déjà traité. Le problème final est *EEPCELL*: étant donné un graphe  $G$ , une surface  $\mathcal{S}$ , un sous-graphe  $H$  de  $G$  plongé cellulièrement sur  $\mathcal{S}$ , décider si il est possible de plonger la totalité de  $G$  sans changer le plongement de  $H$ . Or, B.Mohar [Moh99] a déjà construit un algorithme décidant ce problème.

De plus, dans le même temps, cet algorithme prouve aussi que la plongeabilité d'un graphe sur un 2-complexe est dans **NP**.

**Second algorithme** Le second algorithme se concentre sur des arguments d'ordre plus combinatoire. L'idée est à l'inverse de restreindre et simplifier le graphe à plonger afin de limiter le nombre de possibilités et ensuite construire un algorithme qui essaye de plonger ce graphe plus simple. Au départ, on a aussi un graphe  $G$  que l'on veut plonger sur un 2-complexe  $\mathcal{C}$ . D'abord, on simplifie  $G$  en supprimant des sommets dont le retrait n'influe pas sur le fait que le graphe soit plongeable ou non (*irrelevant vertex method*). Cela permet d'obtenir une décomposition en branche de  $G$  dont la largeur est bornée par la taille de  $\mathcal{C}$ . Ensuite, on présente un algorithme de programmation dynamique qui décide si  $G$  est plongeable sur un complexe en parcourant une décomposition de  $G$ . Si la stratégie utilisée est relativement standard, les arguments et les détails pour appliquer cette stratégie sont spécifique à cet algorithme.

De plus, cet algorithme fonctionnant en particulier pour les surfaces et n'utilisant aucun algorithme précédent décidant si un graphe est plongeable sur une surface, il offre une nouvelle stratégie pour construire un algorithme FPT décidant si un graphe est plongeable sur une surface. La complexité en la taille du graphe est moins bonne que celle d'autres algorithmes précédents pour décider la plongeabilité d'un graphe sur des surfaces. Par exemple, B.Mohar [Moh99] et K.Kawarabayashi, B.Mohar, B.Reed [KMR08] donne tout deux un algorithme linéaire pour décider la plongeabilité d'un graphe sur des surfaces. Cependant, les arguments prouvant la correction de cet algorithme sont plus simples que les arguments des autres méthodes (cf Section 4.1.2).

## 1.4 Organisation du reste de la thèse

Dans le Chapitre [3](#) nous allons commencer par présenter les notions de base nous utiliserons dans le reste de la thèse: topologie, complexité paramétrée, théorie des graphes, des surfaces et des complexes.

Ensuite, dans le Chapitre [4](#), nous allons présenter un état de l'art de problèmes de plongement topologique d'un complexe sur un autre et en particulier une surface ainsi que des problèmes proches des problèmes de plongements topologiques de graphe sur les 2-complexes et les surfaces.

Par la suite, dans le Chapitre [5](#), nous allons d'abord détailler les motivations légitimant l'intérêt du problème de plongements de graphe sur lequel se concentre cette thèse. Dans ce chapitre, nous allons aussi présenter les notions et les propriétés fondamentales sur les 2-complexes que nous avons définies et prouvées que nous utiliserons dans la définition des deux algorithmes.

Après, nous allons présenter respectivement le premier et le second algorithme construits durant cette thèse dans les Chapitres [6](#) et [7](#).

Enfin, nous concluons au Chapitre [8](#) en exposant de futures directions de recherches potentielles en lien avec notre travail.



## Introduction in English

There is a well-known mathematical problem published first by Henry Dudeney [Dud13], the 3-houses problem or 3-utilities problem: there are 3 houses that we want to connect to a electricity plant, a gas pipeline and a water source such the pipes do not cross (see Figure 2.1). This problem is known to have no solution. Trying to solve this problem amounts to try to link 3 points of the plane (the houses) to each of 3 other points of the plan (the utilities) with a path such that these paths do not cross. Mathematically, this set of 6 points such that 3 are linked to the 3 others is known as  $K_{3,3}$ . This way, the problem amounts to looking for a drawing of  $K_{3,3}$  without crossing on the plane.

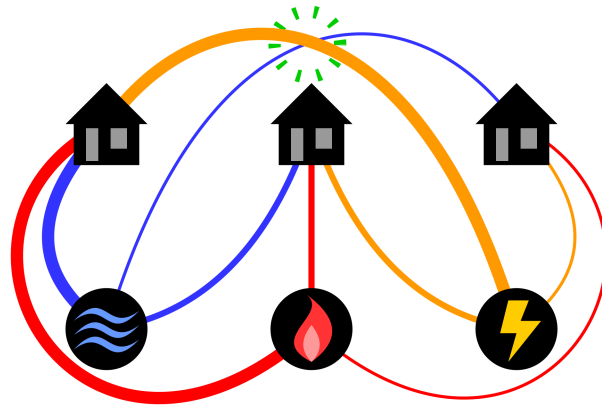


Figure 2.1: An example of the 3-utilities problem. Source: Cmglee

In mathematics, a set of points, called vertices, where some pairs, called edges, are linked is called a graph and a drawing without crossing of a graph is called an embedding. Thus, under innocuous appearances, the 3-houses problem is in fact an iconic case of a classical mathematical problem, graph planarity: given a graph  $G$ , decide whether there exists an embedding of  $G$  on the plane. The absence of a solution for the 3-utilities problem is expressed by saying that  $K_{3,3}$  is not planar. Moreover, the graph planarity

problem can be solved in time linear of the size of the graph [HT74]. Although the strategy used by J.Hopcroft and R.Tarjan is the first working in linear time, the graph planarity problem has been vastly studied [Pat06] and there exist several other strategies to decide the graph planarity problem, some being older [Kur30] other being quicker [FOR06]. Moreover, many refinements very close to the graph planarity problem have been studied. For example, instead of embedding graphs on the plane, we can try to place the vertices of the graph on the vertices of the smallest grid possible such that, for every quadruplet of vertices  $u, v, w, x$  where both pairs of vertices  $\{u, v\}$  and  $\{w, x\}$  are edges of the graph, the segment linking  $u$  to  $v$  does not intersect the segment linking  $w$  to  $x$  [FPP90].

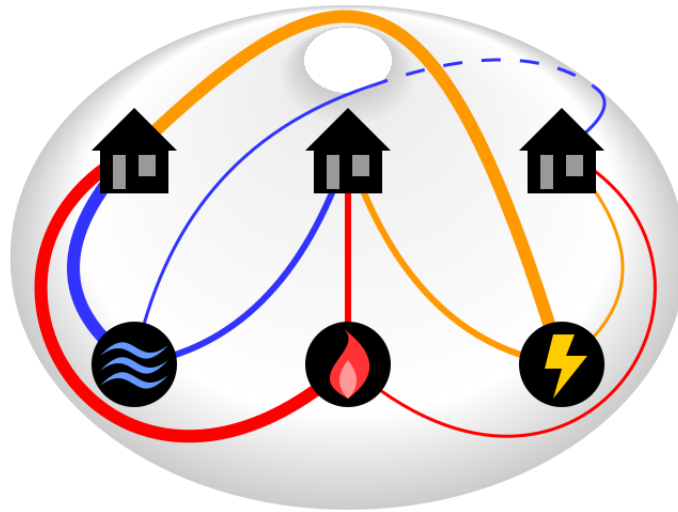


Figure 2.2: A solution of the 3-utilities problem on the torus. Source: [Cmglee]

Similarly, graph planarity is also a particular case of a more general problem, the embeddability of a graph on a surface: given a graph  $G$  and a surface  $\mathcal{S}$ , decide whether there exists an embedding of  $G$  on  $\mathcal{S}$ . The embeddings we consider are topological embeddings, meaning that the edges can be drawn as any curve. For example, although  $K_{3,3}$  is not embeddable on the plane, it can be embedded on the surface of an object with an handle like a torus (Figure 2.2). The embeddability of a graph on a fixed surface can be decided in time linear in the size of the graph. Thereby, B.Mohar [Moh99] and K.Kawarabayashi, B.Mohar et B.Reed [KMR08] both present an algorithm deciding the embeddability of a graph on a surface in linear time when the surface is fixed. Nonetheless, in both cases the complexity of the algorithm rises very quickly when the complexity of the surface, here its number of handles, increases. This explosion of the complexity of the algorithm when the surface becomes more complex is expected because the embeddability of a graph on an arbitrary surface is **NP-hard** [Tho89]. Thus, it is not reasonable to hope for an algorithm deciding the embeddability of a graph on arbitrary surface working in globally polynomial time (otherwise, it would prove that **P=NP**).

When we want to represent a surface in a finite manner, a classical method is to

represent it as a set of triangles pasted together. Nonetheless, it is possible to construct some sets of triangles pasted together that do not correspond to a surface. For example, by pasting three triangles along the same side, we obtain a structure that does not correspond to any surface. Thus, we define a structure called 2-complexes such that these pasting are all a particular case of it. A 2-complex is a graph where some cycles of length 3 of the graph are filled with a triangle (Figure 2.3). The embeddability of a graph on a surface is a particular case of the problem studied in this thesis, **the embeddability of a graph on a 2-complex: given a graph  $G$  and a 2-complex  $\mathcal{C}$ , decide whether there exists an embedding of  $G$  on  $\mathcal{C}$** . Figure 2.4 shows an example of embedding of a graph on a 2-complex.

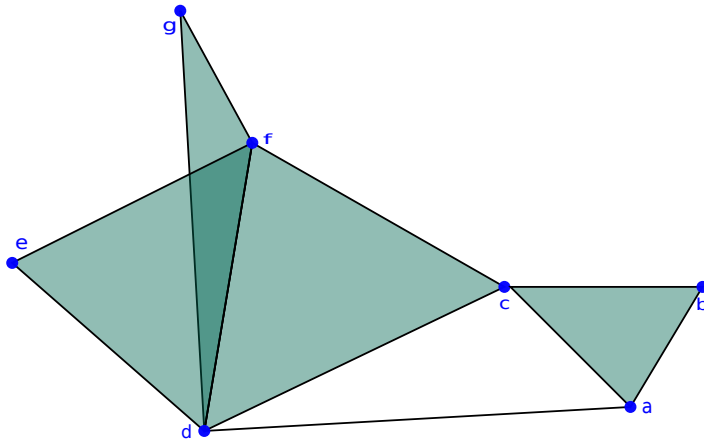


Figure 2.3: a 2-complex with vertices  $\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}$  and  $\{g\}$ , edges  $\{a, b\}, \{a, c\}, \{b, c\}, \{a, d\}, \{d, e\}, \{d, f\}, \{e, f\}, \{d, g\}, \{f, g\}, \{c, f\}$  and  $\{c, d\}$ , and triangles  $\{a, b, c\}, \{d, f, g\}, \{d, e, f\}$  et  $\{c, d, f\}$ .

## 2.1 Motivation

This problem is quite large. For example, we could extend the problem furthermore to 3-complexes by adding to  $\mathcal{C}$  some full tetrahedra bounded by triangles or to complexes of higher dimension by adding to  $\mathcal{C}$  some simplices of dimension 4 or more. All graphs are embeddable on  $\mathcal{C}$  since all graphs are embeddable in a space of dimension 3 or more. So, the embeddability of a graph on a 2-complex is the largest non-trivial case in which we try to embed a graph on a structure formed by vertices, edges, triangles, full tetrahedra and their equivalents of higher dimension. Moreover, there are several classical variations of graph embeddability that can be reduced to the embeddability of a graph on a 2-complex (see Section 5.1.1), in particular the embeddability of a graph on a surface.

Furthermore, classical strategies used to construct an algorithm for embedding on surfaces do not apply to 2-complexes (see Section 5.1.2). Thus, there is no obvious

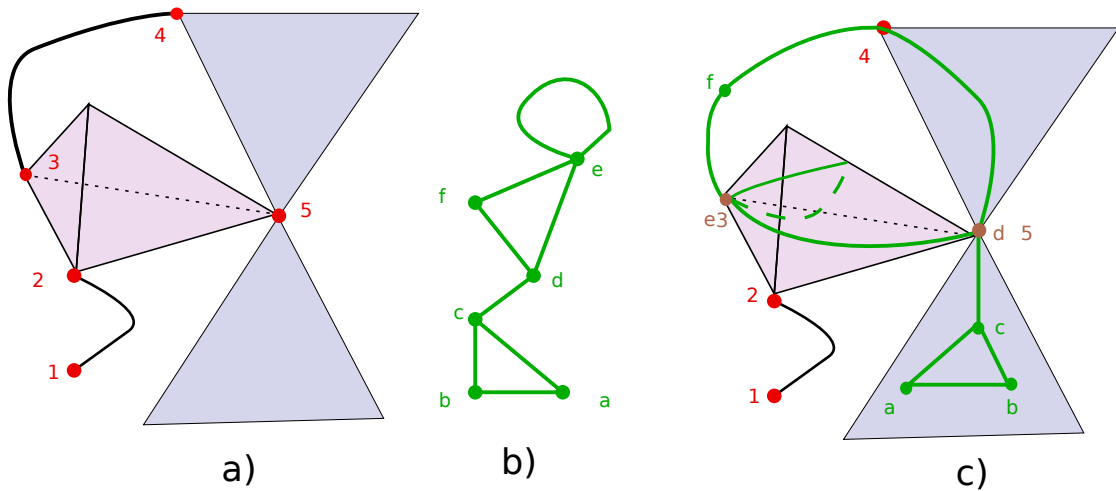


Figure 2.4: a): the 2-complex  $\mathcal{C}$  formed by a sphere up to homeomorphism (a hollow tetrahedron) and two disks sharing one point, one isolated segment between the sphere and one disk and one isolated segment attached to only the sphere. b): a graph  $G$  to embed on it c): an embedding of  $G$  on  $\mathcal{C}$ .

method to decide efficiently whether a graph is embeddable on a 2-complex that can be deduced from the methods for the embeddability of graphs on surfaces.

## 2.2 Related domains

Problems concerning topological embeddings of graphs are key problems of two domains at the interface between computer sciences and mathematics: topological graph theory and computational topology.

**Topological graph theory** The goal of topological graph theory is to study and use graphs seen as a topological object. To do so, we use some topological representations of a graph through an embedding or a drawing of it, most of the time on a surface.

Then, we can use the topological properties of these representations of a graph when studying it. Indeed, for example, we can prove that all planar graphs are 4-colorable [AH76].

Conversely, we can also use graphs to obtain a combinatorial representation of topological spaces and properties. Then, using this combinatorial representation, we can study problems on topological objects and properties with combinatorics. For example, B.Mohar and C.Thomassen [MT01] present basic elements of the analysis of both surfaces and topological embeddings on surfaces seen as combinatorial objects using graphs embedded on the surfaces.

In both cases, it is important to be able to quantify as precisely as possible the complexity of graphs in order to respectively either be able to deduce as much properties

as possible or be able to quantify as precisely as possible the complexity of the space the graph represents.

Moreover, since graphs are often considered embedded on surfaces, it is standard to estimate the complexity of a graph according to the surfaces on which the graph is embeddable. As the set of graphs embeddable on a surface is characterized by the number of handles and Möbius bands composing it (its genus), it is standard to quantify the complexity of a graph with the minimal genus of a surface on which this graph is embeddable. This genus is called the genus of a graph.

**Computational topology** [Veg04] With the development of computer sciences, a domain at the interface of topology and computer science appeared: computational topology, presented as such one of the first times by T.Dey, H.Edelsbrunner and S.Guha [DEG99]. The idea is to try to construct some algorithms that either decide whether a topological property is verified or construct an object satisfying a topological property.

Graph topological embedding problems are a particular case of a key class of problems in computational topology: general embedding problems where the goal is to embed one topological space into another. Thus, there exist some key problems of other domains of computational topology that can be reduced to a case of topological embedding problem. Then, it would allow to use results known for topological embeddings to solve the reduced problem. For example, consider the trivial knot problem which is a central problem in knot theory: given a knot in  $\mathbb{R}^3$ , decide whether this knot is the trivial knot. Since a knot is trivial if and only if there exists a disk embedded in  $\mathbb{R}^3$  such that its boundary is the knot [JT95], the trivial knot problem can be reduced to a topological embedding problem.

Moreover, simplicial complexes are a structure appearing naturally in algorithmic topology. For example, given a topological space  $\mathcal{T}$ , consider a complex  $\mathcal{C}$  embedded bijectively on  $\mathcal{T}$ , called a triangulation of  $\mathcal{T}$ . In algorithmic differential topology, it is common to either look for a triangulation of  $\mathcal{T}$  diffeomorphic to it or use some triangulations of  $\mathcal{T}$  when studying problems on  $\mathcal{T}$  [BM07]. This way, triangulations are an example of the representation of a topological space through a complex showing how these can arise naturally in algorithmic topology.

## 2.3 Contributions of this thesis

This thesis presents two algorithms that both take a graph  $G$  and a 2-complex  $\mathcal{C}$  as an input and decide whether it is possible to embed  $G$  on  $\mathcal{C}$ . We provide the first algorithms solving this problem.

The first algorithm was defined in order to have an algorithm working in polynomial time when the complex is fixed. Nonetheless, in this algorithm, the degree of the polynomial depends on the size of the complex; in technical terms, the algorithm is not FPT (*fixed parameter tractable*) with parameter the size of the complex. The second algorithm is a FPT algorithm in the size of the complex.

The two algorithms differ greatly in the strategy they use to solve the problem.



**First algorithm [de2018embedding]** In the first algorithm, the principle is to construct a sequence of more and more restricted problems where a part of the graph is already embedded. We start with an instance of the problem, a graph  $G$  and a 2-complex  $\mathcal{C}$  where we want to decide whether  $G$  is embeddable on  $\mathcal{C}$ . We will embed on  $\mathcal{C}$  a subgraph  $H$  of  $G$  of bounded size. The idea is to use some topological arguments to restrict more and more the structure of both  $\mathcal{C}$  and the embedding of  $H$  on it even if it means to add some vertices and edges to  $G$ . Eventually we obtain a problem already studied B.Mohar [Moh99], *EEPCELL*: given a graph  $G$ , a surface  $\mathcal{S}$  and subgraph  $H$  of  $G$  cellularly embedded on  $\mathcal{S}$ , decide whether it is possible to embed the whole  $G$  without modifying the embedding of  $H$ .

Moreover, at the same time, this algorithm also proves that the embeddability of a graph on a 2-complex is in **NP**.

**Second algorithm [CM21]** The second algorithm focuses on combinatorial arguments. The idea is to first restrict and simplify the input graph in order to limit the number of possible graphs and then construct an algorithm to embed these simpler graphs. We start with an instance of the problem, a graph  $G$  and a 2-complex  $\mathcal{C}$  where we want to decide whether  $G$  is embeddable on  $\mathcal{C}$ . First, we simplify  $G$  by removing vertices such that their removal does not modify the embeddability of  $G$  (*irrelevant vertex method*). This allow to have a branch decomposition of  $G$  with branchwidth bounded by the size of  $\mathcal{C}$ . Then, we present a dynamic programming algorithm deciding whether  $G$  is embeddable on  $\mathcal{C}$  using a branch decomposition of  $G$ . Although the global strategy is quite standard, the details and arguments to use this strategy are specific to this algorithm.

This algorithm works in particular for surfaces and does not use any previous algorithm deciding the embeddability of graphs on surface. So, it gives a new strategy to construct an FPT algorithm deciding whether a graph is embeddable on a surface. This algorithm is not as efficient as previous algorithms deciding graph embeddability on surfaces. For example, both B.Mohar [Moh99] and K.Kawarabayashi, B.Mohar, B.Reed [KMR08] provides algorithms to decide graph embeddability on surfaces in linear time in the size of the input graph. Nonetheless, the arguments proving its correctness are simpler than the arguments of other methods (see Section 4.1.2).

## 2.4 Organization of the rest of the thesis

In Chapter 3 we will start by presenting the basic notions that we will use in the rest of the thesis: topology, parameterized complexity, graph theory, surfaces and simplicial complexes.

Then, in Chapter 4, we will present a state of the art of topological embedding problems of a simplicial complex on another, in particular on a surface, together with some problems close to topological embeddings of graphs on surfaces and 2-complexes.

Afterwards, in Chapter 5, we will first detail the motivations legitimizing the interest of the problem of embeddability of graphs on 2-complexes. In this chapter we will also

present the fundamental notions and properties on 2-complexes that we have defined and proved that we will use in both algorithms.

Then, we will present the first and the second algorithm we constructed during this thesis in respectively Chapter [6](#) and [7](#).

Eventually, we will conclude in Chapter [8](#) by presenting some future potential research directions linked to the work of this thesis.



# Preliminaries

In this chapter, we will present basic definitions and properties that will be used along the thesis: topology, graph theory, simplicial complexes, surfaces and computational complexity

## 3.1 Topological spaces

In all this thesis we will consider topological spaces.

**Definition 3.1.1** (topological space). A topological space  $X = (\mathbb{E}, \mathbb{U})$  is a pair where  $\mathbb{E}$  is a set and  $\mathbb{U}$  a set of subsets of  $\mathbb{E}$  called the open sets of  $\mathbb{E}$  such that:

- Both  $\emptyset$  and  $\mathbb{E}$  are elements of  $\mathbb{U}$ .
- Any intersection of a finite set of elements of  $\mathbb{U}$  is an element of  $\mathbb{U}$ .
- Any union of a collection (even infinite) of elements of  $\mathbb{U}$  is an element of  $\mathbb{U}$ .

We will suppose known basic elements of topology like continuity, closed and compact sets. We refer to [\[Sti93\]](#) for formal definitions.

### 3.1.1 Homeomorphisms

First, there is a key notion in topology: homeomorphism.

**Definition 3.1.2.** Let  $X$  and  $Y$  be two topological spaces and  $f$  a function from  $X$  to  $Y$ . Then,  $f$  is a homeomorphism (from  $X$  to  $Y$ ) if it is continuous, bijective and its inverse function is also continuous. When  $X=Y$ ,  $f$  is called a self homeomorphism of  $X$ .

The notion of homeomorphism is very important because two homeomorphic spaces have the same topological properties. Moreover, being homeomorphic happens to be an equivalence relation. Thus, when looking for topological properties, spaces are often considered up to homeomorphism. In the rest of this thesis we will often implicitly consider spaces up to homeomorphism.

### 3.1.2 Curves

One key kind of functions in the study of embedding of graphs is curves.

**Definition 3.1.3.** *Let  $X$  be a topological space. Then a curve (on  $X$ ) is a continuous function from  $[0, 1]$  to  $X$ .*

*Let  $\ell$  be a curve on  $X$ . Then,*

- $\ell$  is closed if  $\ell(0) = \ell(1)$ .
- $\ell$  is simple if it is either injective, or closed and injective on  $[0, 1)$ .

Later, a curve will be identified with its image. Moreover, any point  $p$  of  $X$  is a closed curve since it is the image of the constant function always equal to  $p$ .

**Homotopy and contractibility of curves** We will look for continuous deformation of the space changing one curve to another with the same endpoints.

A such deformation is called a homotopy of a curve.

**Definition 3.1.4.** *Let  $X$  be a topological space,  $\lambda$  a continuous function from  $[0, 1]^2$  to  $X$ , and  $\ell$  and  $\ell'$  two curves on  $X$  such that  $\ell(0) = \ell'(0)$  and  $\ell(1) = \ell'(1)$ . Then,  $\lambda$  is a homotopy from  $\ell$  to  $\ell'$  if for all  $x \in [0, 1]$ ,  $\lambda(x, 0) = \ell(x)$ ,  $\lambda(x, 1) = \ell'(x)$  and for all  $t \in [0, 1]$ ,  $\lambda(0, t) = \ell(0)$  and  $\lambda(1, t) = \ell(1)$ . Moreover, two curves with the same endpoints are homotopic if there is a homotopy from one to the other.*

Intuitively, the second parameter represents the time along the deformation. This way, consider a homotopy  $\lambda$  between two curves  $\ell$  and  $\ell'$ , then for all  $t$ , the function  $x \rightarrow \lambda(x, t)$  is the state at the instant  $t$  of the deformation from  $\ell$  to  $\ell'$ .

Moreover, in this thesis, homotopy will mainly be used to define a key notion on curves: *contractibility*.

**Definition 3.1.5.** *Let  $\ell$  be a closed curve on a topological space  $X$ . Then  $\ell$  is contractible if it is homotopic to a point of  $X$ .*

## 3.2 Graphs

Informally, a graph is a set of vertices connected by undirected edges. More formally there are two classical definitions for graphs, one broad later called a *multigraph* and one more restrictive called *simple graphs*.

### 3.2.1 Multigraphs and simple graphs

The most general definition for a graph in this thesis, called a *multigraph*, is the following:

**Definition 3.2.1.** *A multigraph is a triple  $G = (V, E, \text{extr})$  where:*

- $V$  is a finite set whose elements are called the vertices of  $G$ .

- $E$  is a finite set whose elements are called the edges of  $G$ .
- $extr$  is a mapping from elements of  $E$  to subset of  $V$  of cardinality 1 or 2.

Let  $e$  be an edge of  $G$ . Then, the elements of  $f(e)$  are called the endpoints of  $e$ . Moreover,  $e$  is a loop if it has only one endpoint.

Often, a more restrictive definition of a graph, called a *simple graph* is used.

**Definition 3.2.2.** A multigraph  $(V, E, extr)$  is a simple graph if  $E$  contains neither a loop nor two edges with the same endpoints.

In a simple graph, an edge is characterized by the pair of its endpoints. Thus, in a simple graph, each edge can be identified with the pair of its endpoints. This way, in a simple graph  $G = (V, E, extr)$ ,  $E$  is a set of pairs of elements of  $V$  and  $extr$  is simply the identity map. Eventually,  $G$  is simply represented as the couple  $(V, E)$  since  $extr$  is obvious.

### 3.2.2 Graph embeddings

When looking at a graph, we can try to draw it on a topological space by drawing vertices as points and edges as curves between their endpoints. More specifically, we can look for a crossing free drawing: an *embedding*.

**Definition 3.2.3.** Let  $G = (V, E, extr)$  be a multigraph and  $X$  a topological space. Then an embedding of  $G$  on  $X$  is specified by an injective function  $f$  from  $V$  to  $X$  and a family of simple curves  $(g_e)_{e \in E}$  such that:

- For all  $e \in E$  with endpoints  $u$  and  $v$  (with  $u = v$  if  $e$  is a loop),  $\{g_e(0), g_e(1)\} = \{f(u), f(v)\}$ .
- For all  $e \in E$  and all  $t \in (0, 1)$ ,  $g_e(t)$  neither lies on the image of  $f$  nor the image of any  $g_{e'}$  for any  $e' \neq e$ .

Let  $\Pi$  be an embedding of  $G$  on  $X$  of image  $Y$ . A face of  $\Pi$  is a connected component of  $X \setminus Y$ .

**Ambient isotopy of an embedding** Sometimes it can be useful to continuously deform one embedding into another. The notion used to represent a such continuous deformation is called an ambient isotopy.

**Definition 3.2.4.** Let  $G$  be a graph,  $X$  a topological space, and  $f$  and  $g$  two embeddings of  $G$  on  $X$ . Then, an ambient isotopy from  $f$  to  $g$  is a continuous family  $(h_t)_{t \in [0,1]}$  of self-homeomorphisms of  $X$  such that  $h_0$  is the identity and  $h_1 \circ f = g$ .

We will use ambient isotopies in particular in Chapter 6 in order to modify embeddings while maintaining some specific properties that do not translate through homeomorphism.

### 3.2.3 New graphs obtained from an old one

When studying a graph  $G$ , it can be useful to look at some graphs associated with  $G$ , often linked to  $G$  by the relation of being a subgraph. Thus, there are some standard constructions of subgraphs that are very useful. Moreover, a minor of a graph is a more general notion of smaller graphs which is sometimes used instead of subgraphs where more adapted.

**Definition 3.2.5.** Let  $G = (V, E, \text{extr})$  and  $G' = (V', E', \text{extr}')$  be two graphs,  $u_0$  a vertex of  $G$ ,  $e$  an edge of  $G$ ,  $V_0$  a subset of  $V$  and  $E_0$  is a subset of  $E$ .

1. Subgraph.  $G'$  is a subgraph of  $G$  (noted  $G' \subseteq G$ ) if  $V' \subseteq V$ ,  $E' \subseteq E$  and  $\text{extr}'$  is the restriction of  $\text{extr}$  to  $E'$ . The notion of subgraph is often the notion used as "smaller graphs" since there is a lot of properties of graphs that propagates to any of its subgraphs.
2. Induced graph.  $G'$  is the graph induced by  $V_0$  if  $V' = V_0$ ,  $E' = \left\{ e \mid e \in E \wedge \text{extr}(e) \subseteq V_0 \right\}$  and  $\text{extr}'$  is the restriction of  $\text{extr}$  to  $E'$ . Similarly,  $G'$  is the graph induced by  $E_0$  if  $E' = E_0$ ,  $\text{extr}'$  is the restriction of  $\text{extr}$  to  $E'$  and  $V' = \left\{ u \mid \exists e \in E_0, u \in \text{extr}'(e) \right\}$ . The graph induced by  $V_0$  is the biggest subgraph of  $G$  whose set of vertices is  $V_0$  while the graph induced by  $E_0$  is the smallest subgraph of  $G$  whose edges are  $E_0$ .
3. Graph Difference. The difference between  $G$  and  $V_0$  (noted  $G \setminus V_0$ ) is the graph induced by  $V \setminus V_0$ . This way we obtain the biggest subgraph without some vertices. Similarly, the difference between  $G$  and  $H$  (noted  $G \setminus G'$ ) is the graph induced by  $E \setminus E'$ . This way, we obtain the complement of a subgraph in a graph.
4. Subdivision.  $G'$  is an atomic subdivision of  $G$  if there exists  $e \in E$  with endpoints  $u$  and  $v$  (with  $u = v$  if  $e$  is a loop) and  $w \in V'$  such that  $V' = V \cup \{w\}$  and  $E' = \{e', e''\} \cup E \setminus e$  such that  $\text{extr}'(e') = \{u, w\}$  and  $\text{extr}'(e'') = \{v, w\}$ . Moreover,  $G'$  is a subdivision of  $G$  if there exists a set of graphs  $G_0, \dots, G_n$  such that  $G_0 = G$ ,  $G_n = G'$  and for all  $0 \leq i < n$ ,  $G_{i+1}$  is an atomic subdivision of  $G_i$ . Subdivisions are useful to "refine" edges of a graph in order to modify edges of a graph by acting on vertices while not having any influence on the image of an embedding of it.
5. Edge contraction.  $G'$  is an edge contraction of  $G$  if there is an edge  $e$  of  $G$  with endpoints  $u$  and  $v$  such that  $G'$  is the graph obtained from  $G$  by removing  $e$  and identifying  $u$  and  $v$ . Thus, there is a vertex  $w \notin V$  such that  $V' = (V \setminus \{u, v\}) \cup \{w\}$ ,  $E' = E \setminus \{e\}$  and  $\text{extr}'$  is the function  $\text{extr}$  except that each instance of either  $u$  or  $v$  in the set of the image of an edge is replaced by  $w$ . Edge contractions are mainly used to define the minors of  $G$ .
6. Minor.  $G'$  is a minor of  $G$  if there exists a sequence of graphs  $G_0, \dots, G_n$  such that:

- $G_0$  is a subgraph of  $G$  and  $G_n = G'$
- for all  $0 \leq i < n$   $G_{i+1}$  is an edge contraction of  $G_i$ .

All subgraphs of  $G$  are also a minor of it. Moreover, minors are sometimes used instead of subgraphs as "smaller graphs" when the properties studied propagate to all minors and not only subgraphs.

### 3.2.4 Graph structures

When studying graphs, it is possible to look at either structures of subgraphs or classes of graphs with a particular structure.

**Paths and cycles** Let  $G$  be a graph and  $u$  and  $v$  two vertices of  $G$ .

A path of  $G$  linking  $u$  and  $v$  is an alternating sequence of vertices and edges of  $G$ :  $v_1, e_1, v_2, \dots, e_n, v_{n+1}$  such that  $v_1 = u$ ,  $v_{n+1} = v$  and for all  $1 \leq k \leq n$   $v_k$  and  $v_{k+1}$  are the endpoints of  $e_k$ .

- A path  $v_1, e_1, \dots, v_{n+1}$  is *simple* if for all  $1 \leq k \neq k' \leq n$ ,  $v_k \neq v_{k'}$ .
- A path  $P$  is a *closed path* of  $G$  if there is a vertex  $u$  of  $G$  such that  $P$  is a path of  $G$  from  $u$  to itself.
- A simple closed path of  $G$  is called a *cycle*

Since a cycle is a simple path, each edge or vertex appears at most once in the cycle. Thus, a cycle immediately induces a cyclic order of both the edges and the vertices appearing in it.

Later, a cycle will be simply defined by the cyclic order of edges appearing in it. Thus, two cycles inducing the same cyclic order of edges will be identified. This cyclic order does not pinpoint a start to the cycle but, except for cycle composed of two vertices, it still implies a direction for the course of this cycle.

**Trees** There is a class of graph widely studied: trees.

**Definition 3.2.6.** A graph  $T$  is a (unrooted) tree if it is both connected and contains no cycle.

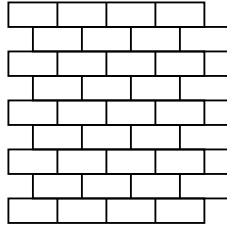
Then, a vertex  $u$  is either a leaf when its degree is 1 or an internal node when its degree is greater than one.

Trees have several key properties linked to its definition.

**Proposition 3.2.7.** Let  $T$  be a tree. Then:

- For each pair of vertices  $u, v$  of  $T$ , there is exactly one simple path of  $T$  between  $u$  and  $v$ .
- For each edge  $e$  of  $T$ ,  $T \setminus e$  is disconnected



Figure 3.1: A  $(10 \times 10)$ -wall.

- Conversely, for each pair of vertices  $u, v$  of  $T$  such that the edge  $\{u, v\}$  is not in  $T$ , the graph obtained from  $T$  by adding  $e$  contains a cycle.

Often, when looking at a tree, we choose one vertex to be the root of the tree. Then, the tree is a *rooted tree*.

In a rooted tree, we can define the descendants of an internal node. Let  $t$  be an internal node of a rooted tree  $T$ . Then, a node  $t'$  is a descendant of  $t$  if  $t$  appears along the path between  $t'$  and the root of  $T$ . Moreover, a node  $t'$  is a child of  $t$  if it is a descendant of  $t$  connected to it.

### Grids and walls

**Definition 3.2.8.** Let  $g$  be an integer. The  $g \times g$ -grid is the graph with vertices  $\{(i, j) | 1 \leq i, j \leq g\}$  and an edge between  $(i, j)$  and  $(i', j')$  if either  $i' = i + 1$  and  $j' = j$  or  $i' = i$  and  $j' = j + 1$ .

Furthermore, a graph  $G$  is a wall of size  $k \times k$  if it is a subgraph of the  $(k \times k)$  grid obtained by removing alternatively the vertical edges of even (resp. odd)  $x$ -coordinate in each even (resp. odd) line, and then the degree-one vertices; see Figure [3.1](#)

## 3.3 Simplicial complexes

In this section, we will present the basic elements of simplicial complexes that will be used in the rest of the thesis. We refer to [Mat03](#) for a broader and more exhaustive presentation of simplicial complexes and their topology.

### 3.3.1 Simplex

Complexes will be defined as a set of canonical bricks: simplices.

**Definition 3.3.1.** A simplex  $S$  is a finite non-empty set whose elements are called the vertices of  $S$ .

The dimension of  $S$  is  $\dim(S) = |S| - 1$ .

Moreover, a simplex  $S'$  is a face of  $S$  if  $S' \subset S$ .

A simplex of dimension  $k > 0$  is often used as a combinatorial definition of a  $k$ -dimensional ball while a simplex of dimension 0 is seen as a point.

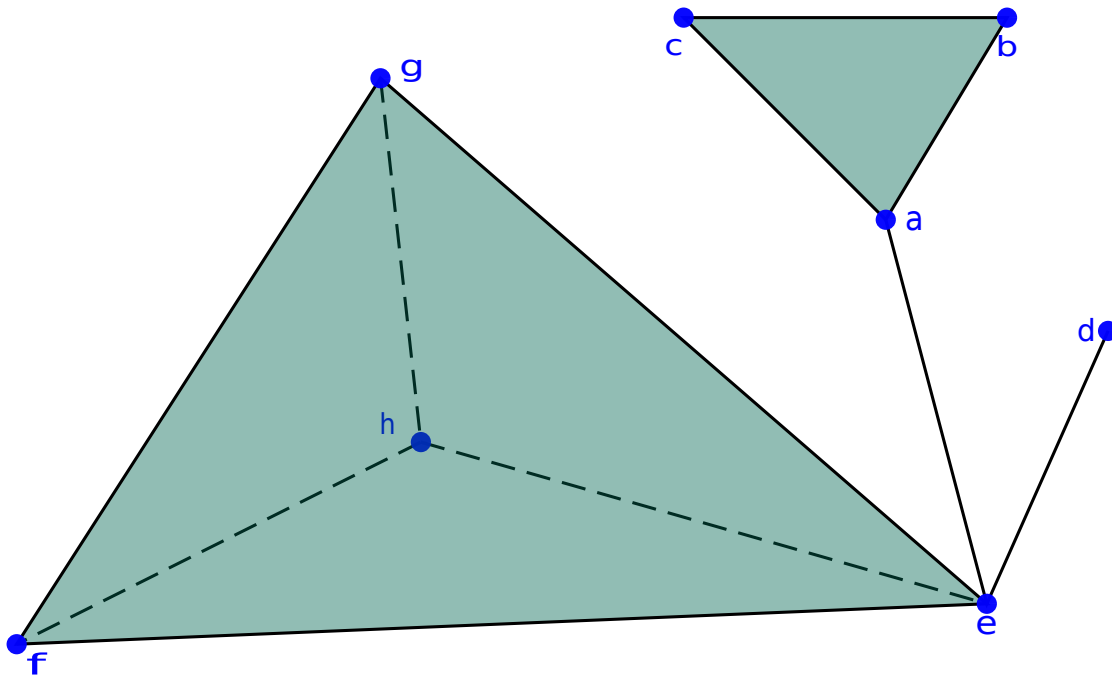


Figure 3.2: A 2-complex with simplices:  $\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}, \{g\}, \{h\}$  of dimension 0 (vertices),  $\{a, b\}, \{a, c\}, \{b, c\}, \{a, e\}, \{d, e\}, \{e, f\}, \{e, g\}, \{e, h\}, \{f, g\}, \{f, h\}, \{g, h\}$  of dimension 1 (edges) and  $\{a, b, c\}, \{e, f, g\}, \{e, f, h\}, \{e, g, h\}, \{f, g, h\}$  of dimension 2 (triangles).

### 3.3.2 Combinatorial definition of a simplicial complex

A simplicial complex is a set of simplices closed under inclusion.

**Definition 3.3.2.** A finite set of simplices  $\mathcal{C}$  is a simplicial complex if, for any simplex  $S$  in  $\mathcal{C}$  and any  $S' \subset S$ , the set  $S'$  is also a simplex of  $\mathcal{C}$ . In other words, a set of simplices is a simplicial complex if it contains all faces of any simplex included in it.

The vertices of  $\mathcal{C}$  are the vertices of any simplex of  $\mathcal{C}$ . Due to the definition of a simplicial complex, the set of vertices of  $\mathcal{C}$  is also the union of the singletons, the simplices of dimension 0, of  $\mathcal{C}$ .

Moreover, let  $k$  be an integer. Then  $\mathcal{C}$  is a  $k$ -dimensional complex (or  $k$ -complex) if the maximal dimension of a simplex in  $\mathcal{C}$  is at most  $k$ .

See Figure [3.2](#) for an example of simplicial complex.

By definition, simple graphs are exactly 1-complexes.

**1-skeleton** Let  $d < k$  be two integers and  $\mathcal{C}$  be a  $k$ -dimensional complex. Then the  $d$ -skeleton of  $\mathcal{C}$  is the simplicial complex  $\mathcal{C}'$  formed by the simplices of dimension at most  $d$  of  $\mathcal{C}$ .

$\mathcal{C}'$  can be a very useful tool in the study of  $\mathcal{C}$ . Sometimes, it allows the use of results and strategies applying to simplicial complexes of dimension  $d$  in the study of  $\mathcal{C}$ . Moreover,  $\mathcal{C}'$  forms the "frame" that host higher-dimensional simplices of  $\mathcal{C}$ . Thus,  $\mathcal{C}'$  can be used instead of  $\mathcal{C}$  when it retains enough of the structure of  $\mathcal{C}$  for the problem considered.

In the rest of this thesis, we will only use the 1-skeleton of a 2-complex  $\mathcal{C}$ , the graph formed by the vertices and edges of  $\mathcal{C}$ .

### 3.3.3 Associated topological space

We can associate a topological space to any simplicial complex called the *topological representation* of  $\mathcal{C}$ .

**Definition** Let  $\mathcal{C}$  be a simplicial complex.

We construct the *topological representation* of  $\mathcal{C}$  by associating inductively a topological space to each simplex of  $\mathcal{C}$ .

First, we associate each vertex to a different point.

Then, let  $S$  be a simplex of  $\mathcal{C}$  of dimension  $k > 0$  such that a topological space has already been associated with the faces of  $S$ . We will always have that the union of the topological space of the faces of  $S$  is homeomorphic to the sphere of dimension  $k$ . We associate to  $S$  a topological space homeomorphic to a ball of dimension  $k$  whose boundary is the union of the topological representation of all faces of  $S$ .

Eventually, the final topological space is the *topological representation* of  $\mathcal{C}$ . It is evident that all topological representation of  $\mathcal{C}$  are homeomorphic. Thus, later, a simplicial complex will be implicitly associated with any of its topological representations.

**The 1-dimensional case** Embeddings of graphs gives a classical way to construct a topological representation of graphs. Thus, an embedding associate a point to each vertex and a space homeomorphic to a simple curve (the ball of dimension 1) whose boundary is the image of vertices to edges.

**Topological spaces associated with 2-dimensional complexes** In this thesis, we will focus on 2-complexes. Let  $\mathcal{C}$  be a 2-complex. As a 2-complex, the simplices composing  $\mathcal{C}$  are vertices, edges and triangles. Thus, the topological space associated with  $\mathcal{C}$  will be a set of closed disks, corresponding to triangles, some sharing a point or a part of their boundary, some isolated arcs, corresponding to edges incident to no triangles, and some isolated points, corresponding to vertices incident to no edge. Alternatively, it is obtained from a simple graph by gluing a disk to some of its cycles of length three.

## 3.4 Surfaces

There is one class of topological representation of 2-complexes where the embedding of graphs has been vastly studied: surfaces.

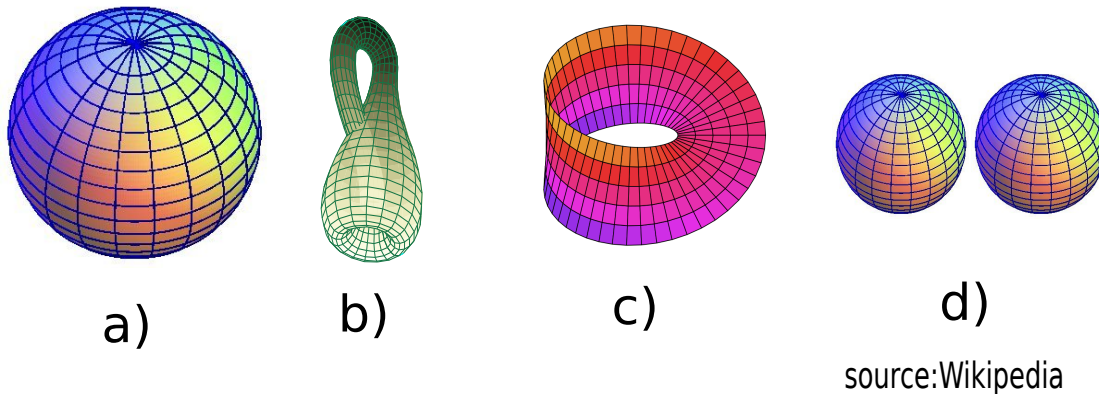


Figure 3.3: a) An orientable surface: a sphere. b) An non-orientable surface: the Klein-bottle. c) A surface with a boundary: the Möbius band. d) A non-connected surface: two spheres.

In this section, we will present the basic elements of surfaces theory that will be used in the rest of the thesis. We refer to [MT01] for a broader and more exhaustive presentation of embeddings on surfaces.

### 3.4.1 Definition

Definitions of surfaces come in several forms, depending on whether the surface is required to be connected and whether it can have boundaries. In this thesis we will define both notions of connected surface without boundary and the notion of surface possibly disconnected, possibly with boundary.

#### Definition 3.4.1. *Definition of a surface possibly disconnected, possibly with boundary*

Let  $\mathcal{S}$  be a compact topological space.  $\mathcal{S}$  is a surface possibly disconnected, possibly with boundary if each point  $p$  of  $\mathcal{S}$  admits a neighborhood of  $p$  homeomorphic to a closed disk. See Figure 3.3 for examples of general surfaces.

Moreover, let  $\mathcal{S}$  be such a surface. A point of  $\mathcal{S}$  is a boundary point of  $\mathcal{S}$  if it admits no neighborhood homeomorphic to an open disk. A connected component of the set of boundary points of  $\mathcal{S}$  is a boundary component of  $\mathcal{S}$ . Moreover, any boundary of a general surface is always homeomorphic to a simple closed curve.

Then, we define surfaces which happen to be exactly connected surfaces with no boundary.

#### Definition 3.4.2. *Surface*

Let  $\mathcal{S}$  be a surface possibly disconnected, possibly with boundary.  $\mathcal{S}$  is a surface if it is connected and without boundary

See a) and b) of Figure 3.3 for examples of surfaces.

Remark: Thus, our definition of surface is the most restrictive one. In particular, a surface possibly disconnected, possibly with boundary, is not necessarily a surface.

### 3.4.2 Facial walks and cellular embeddings

Embeddings on surfaces have a particular structure that allow to define some objects specific to it. First, we can define cellular embeddings where the faces are all disks. Then, we can define the closed paths formed by going along a specified face of the embedding: the facial walks.

**Cellular embeddings** There is one particular case of embeddings on surfaces called *cellular embeddings* where faces are disks. An embedding  $\Pi$  on a surface possibly disconnected without boundary  $\mathcal{S}$  is a *cellular embedding* if each face of  $\Pi$  is homeomorphic to an open disk.

Cellular embeddings are a key tool for the study of surfaces. In particular, they allow to define the two significant characteristics of it: genus and orientability (See Section [3.4.3](#)).

**Facial walks** A facial walk of a face of an embedded graph represents the part of the graph that bounds this face.

**Definition 3.4.3.** *Let  $\Pi$  be an embedding of a graph  $G$  embedded on a surface  $\mathcal{S}$ . Consider a face  $F$  of  $\Pi$ . Then, the frontier of  $F$  is composed of one or more closed curves corresponding to parts of  $\Pi$ . Consider one such curve  $\ell$ . Then, when going along  $\ell$ , we meet alternatively the image of vertices and edges of  $G$ . This way, we obtain a closed path of  $G$  called a facial walk of the face  $F$  of  $\Pi$  (associated with  $\ell$ ).*

*Remark: It is possible that the curve considered is a single point. Then, the corresponding facial walk is composed of a single vertex, the vertex mapped on this point.*

### 3.4.3 Genus-orientation characterization of a surface

Up to homeomorphism, a surface only depends on two of its characteristics: its orientability and its genus.

**Orientability** The orientability of a surface can be defined with a cellular embedding of a graph on it.

**Definition 3.4.4.** *Let  $\mathcal{S}$  be a surface and  $\Pi$  a cellular embedding of a graph on  $\mathcal{S}$ .*

*$\Pi$  is orientable if it is possible to choose a direction around the perimeter of each face  $F$  of  $\Pi$ , associating a direction to each appearance of an edge along the facial walk of  $F$ , such that every appearance of an edge in facial walks of  $\Pi$  are in opposite direction. See Figure [3.4](#) for an example of orientation of a cellular embedding on the sphere.*

*If one cellular embedding on  $\mathcal{S}$  is orientable then all are. Thus, it is possible to define a surface as orientable if it admits an orientable cellular embedding.*

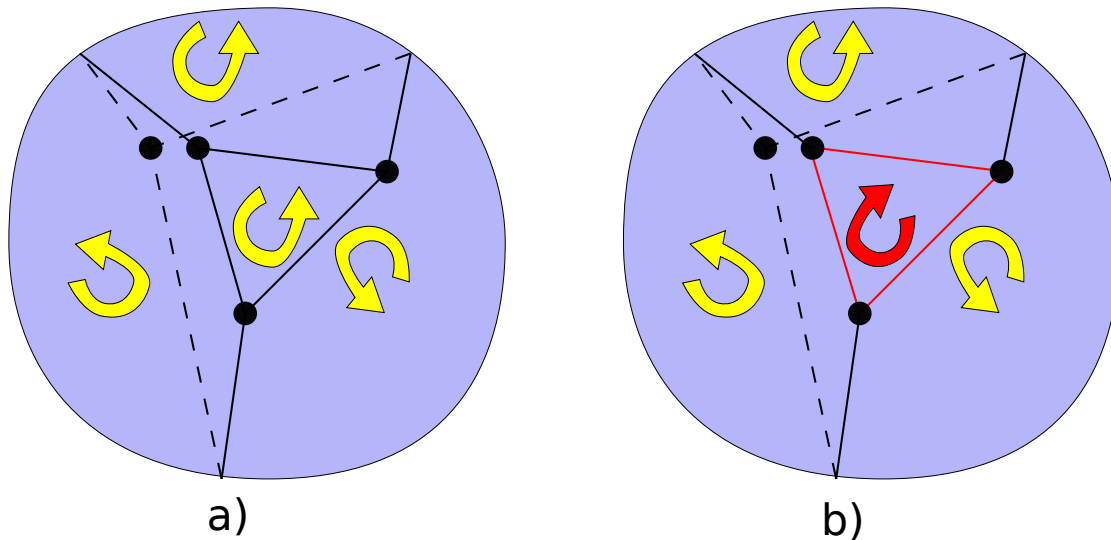


Figure 3.4: a) A valid orientation on the sphere b) An invalid orientation of the sphere.

*Intuitively, an orientable surface is a surface with an inside and outside. See Figure 3.3 for examples of general surfaces both orientable and non-orientable.*

Closed curves on orientable surfaces have one key property that can be used as an alternative definition of orientable surface depending their number of sides.

**Definition 3.4.5. One- and two-sided curves** Let  $\mathcal{S}$  be a surface and  $\ell$  be a simple closed curve. Then,  $\ell$  is two-sided if there exists a connected neighborhood  $N$  of  $\ell$  such that  $N \setminus \ell$  is not connected. Otherwise, if no connected neighborhood of  $\ell$  is disconnected by it,  $\ell$  is one-sided.

Then, we have the following:

**Proposition 3.4.6.** Let  $\mathcal{S}$  be a surface. Then,  $\mathcal{S}$  is orientable if and only if all simple closed curves are two-sided.

Thus, on a surface, there is two kinds of simple closed curves: the ones admitting a connected neighborhood separated by it that are said to be two-sided and the one that does not separate any of their connected neighborhoods. As stated by the previous proposition, all simple closed curves on a surface are two-sided if and only if this surface is orientable. So, one-sided simple closed curves only appears on non-orientable surface.

**Genus of a surface** Let  $\mathcal{S}$  be a connected surface with  $b$  boundaries,  $G$  a graph with  $n$  vertices and  $e$  edges and  $\Pi$  a cellular embedding of  $G$  on it with  $k$  faces. We define the *Euler characteristic (of  $\mathcal{S}$ )* as  $\chi = n - e + k$ . Then, the Euler characteristic have two key properties:

- First, the Euler characteristic is the same for all cellular embeddings on  $\mathcal{S}$ .

- Second, for all connected general surfaces with boundary, its Euler's characteristic  $\chi$  satisfies  $\chi \leq 2$ .

Then the non-negative integer  $g = 2 - \chi$  is called the (*general*) *genus* of  $\mathcal{S}$ . If  $\mathcal{S}$  is orientable then  $g$  is always even and the *orientable genus* of  $\mathcal{S}$  is  $\frac{g}{2}$ .

**Euler's theorem** Euler's theorem proves that orientability and genus characterize a surface up to homeomorphism.

**Theorem 3.4.7.** *Euler's theorem* Let  $\mathcal{S}$  and  $\mathcal{S}'$  be two surfaces. Then,  $\mathcal{S}$  and  $\mathcal{S}'$  are homeomorphic if and only if they 1) have both the same genus and 2) they are either both orientable or both not orientable.

Then, it obviously implies that the genus and orientability of a surface characterize the set of graphs embeddable on it.

**Corollary 3.4.8.** *Let  $\mathcal{S}$  and  $\mathcal{S}'$  be two surfaces. If 1)  $\mathcal{S}$  and  $\mathcal{S}'$  have both the same genus and 2) they are either both orientable or both not orientable, then the set of graphs embeddable on  $\mathcal{S}$  is the same as the set of graphs embeddable on  $\mathcal{S}'$ .*

**Extension to non-connected surface and surface with boundary** Both notions of genus and orientability can be extended to surface possibly disconnected, possibly with boundary.

First, the orientability is defined identically as the definition of orientability for surfaces without boundaries by considering an embedding where all faces are homeomorphic to a disk. For non-connected surfaces, we say that it is orientable if all of its connected components are orientable.

Then, we also define the Euler's characteristic of a surface with boundary  $\mathcal{S}$  from a cellular embedding on it with the same formula,  $\chi = n - e + k$  where  $n$  and  $e$  is respectively the number of vertices and the number of edges of the embedded graph and  $k$  the number of faces of the embedding. Nonetheless, the genus is then defined as  $g = 2 - b - \chi$  where  $b$  is the number of boundaries of  $\mathcal{S}$ . This way we still have that  $g \geq 0$ . Moreover,  $g$  is also the genus of the surface without boundary obtained from  $\mathcal{S}$  by, for each boundary component  $B$ , adding a closed disk whose boundary is identified with  $B$ . For non-connected surfaces, we define its genus as the sum of the genus of each of its connected components.

**Genus of a graph** Let  $G$  be a graph. We can associate to  $G$  the minimal genus of a surface on which  $G$  is embeddable. Thus, the *genus* of  $G$  is  $g$  if there exists a surface  $\mathcal{S}$  of genus  $g$  such that  $G$  is embeddable on  $\mathcal{S}$  and  $G$  is not embeddable on any surface of genus less than  $g$ . Similarly, the *non-orientable genus* of  $G$  is  $g$  if there exists a non-orientable surface  $\mathcal{S}$  of genus  $g$  such that  $G$  is embeddable on  $\mathcal{S}$  and  $G$  is not embeddable on any non-orientable surface of genus less than  $g$ . Moreover, the *orientable genus* of  $G$  is  $g$  if there exists an orientable surface  $\mathcal{S}$  of orientable genus  $g$  such that  $G$  is embeddable on  $\mathcal{S}$  and  $G$  is not embeddable on any orientable surface of orientable genus less than  $g$ .

### 3.4.4 Combinatorial representation of an embedding on a surface

In this thesis, we are only looking at embeddings up to homeomorphism. Then, it is natural, instead of giving its exact image, to look for a way to represent an embedding such that all homeomorphic embeddings have the same. This will be done for cellular embeddings by giving a representation that characterizes the facial walks of the embedding. Thus, it is obvious that two embeddings  $\Pi$  and  $\Pi'$  of a graph  $G$  which faces both are homeomorphic and admits the same facial walks are homeomorphic. Then, since in a cellular embedding all faces are homeomorphic to disks, knowing the facial walks of a cellular embedding characterizes both the surface on which the graph is embedded and the way it is embedded.

There is a way to represent these facial walks and thus a cellular embedding up to homeomorphism: the combinatorial map. Combinatorial maps will not be defined in this thesis. We refer to [MT01, Section 3.3] for a formal definition of the way to associate an embedding to a combinatorial map. Here, we will only define the rotation systems and combinatorial maps while giving the intuition of their use to represent cellular embeddings on surfaces.

**Rotation system** First, we define the rotation system for embeddings on orientable surfaces.

Let  $G$  be a graph,  $\mathcal{S}$  an orientable surface and  $\Pi$  a cellular embedding of  $G$  on  $\mathcal{S}$ . Then  $\Pi$  maps the edges incident to a vertex around it in a cyclic order, which is well defined if an orientation of  $\mathcal{S}$  is chosen. Thus,  $\Pi$  induces naturally a map that associates to each vertex a cyclic order of the edges incident to it. This map is a *rotation system* of  $\Pi$ .

Since the rotation system is a map from vertices of  $G$  to a cyclic order of the edges incident to it, we can define a rotation system without any underlying embedding.

**Definition 3.4.9.** *Let  $G$  be a graph. A rotation system of  $G$  is a map that associates to each vertex of  $G$  a cyclic order of the edges incident to it.*

**Reconstructing facial walk from a rotation system in the orientable case** A rotation system of a cellular embedding on an orientable surface (connected without boundary) suffices to reconstruct all facial walks of it:

Let  $G$  be a graph,  $\mathcal{S}$  an orientable surface,  $\Pi$  a cellular embedding of  $G$  on  $\mathcal{S}$  and  $\Lambda$  the rotation system of  $\Pi$ .

The reconstruction of facial walks of faces of  $\Pi$  from  $\Lambda$  is as follows:

First, choose one vertex  $u$  and one edge  $e$  incident to it.

Then, until the sequence  $u, e$  appears a second time in the facial walk, repeat the following step:

Let  $v, e'$  be the end of the sequence already constructed with  $v$  a vertex and  $e'$  an edge incident to it. Let  $v'$  be the other end of  $e'$  and  $e''$  be the edge following  $e'$  in the cyclic order  $\Lambda(v')$ . Add  $v', e''$  to the facial walk.



**The issue of non-orientable surface** If  $\mathcal{S}$  is not orientable, the same construction strategy does not work. We will not define formally rotation systems for non-orientable surfaces, but we only indicate intuitively why more information is required. Since the surface is non-orientable, the cyclic ordering of the edges around each vertex is defined up to reversal only.

For example, see Figure 3.5. Both embeddings of  $G$  seem to have the same rotation system  $\Lambda$ :

- A: 1,2,3
- B: 2,4,5
- C: 1,3
- D: 4,5

Nonetheless, if we go along the edges 3 and 1 on the Möbius band, we see that we went from the yellow side of A to its green one. Thus, the next vertex seems to be 3 again and not 2. This way, after going through the edge 1, the cyclic order around each vertex looks reversed.

So, on non-orientable surfaces, we can have some edges that are "twisted" (here it is 1) and going through these reverse all cyclic order of the rotation system afterwards. Thus, when representing an embedding on a non-orientable surface we need more information than simply a rotation map, we also need to know for each edge whether it keeps the cyclic orders around vertices or not.

To do so, we add to each edge a signature: positive when there is no inversion of the cyclic order and negative otherwise.

**Combinatorial map of a cellular embedding** This leads to the definition of a combinatorial map.

**Definition 3.4.10.** *Let  $G$  be a graph. Then, a combinatorial map of  $G$  is a couple  $(\Lambda, \sigma)$  where  $\Lambda$  is a rotation system of  $G$  and  $\sigma$  is a map from the edges of  $G$  to  $\{-1, 1\}$  called the signature of the combinatorial map.*

We can reconstruct the facial walks from a combinatorial map like the orientable case except that we reverse all orders in the rotation system when we add an edge with signature  $-1$ .

### 3.5 Fixed parameter tractability

In this section, we will present the basic elements of parameterized complexity that will be used in the rest of the thesis. We refer to [Cyg+15] for a broader and more exhaustive presentation of the domain of parameterized complexity.

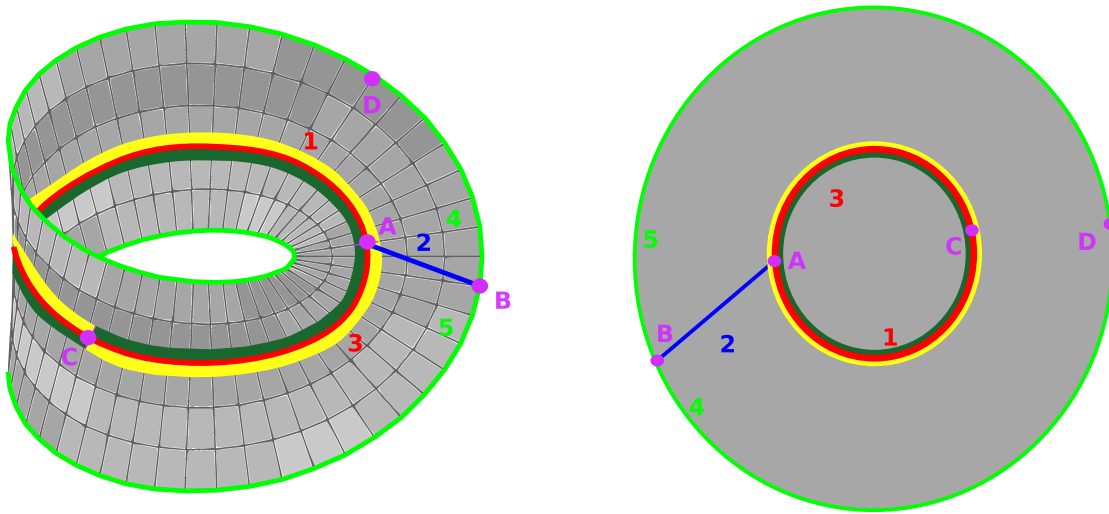


Figure 3.5: Two cellular embeddings of a graph  $G$  on respectively a Möbius band and a disk with the same rotation system

### 3.5.1 Fixed parameter tractability: definition and motivation

**Problems** In this thesis we will restrict ourselves to decision problems. A *decision problem*  $P$  is a function that associate inputs to either true or false. Then, an algorithm solves  $P$  if for each input  $I$  it returns  $P(I)$ .

For example, we can define the PRIMALITY problem: given an integer  $n$ , decide whether  $n$  is a prime number.

**P and NP classes** There are classic classes of decision problems:

- **P**. A problem  $P$  is in **P** if there exists a constant  $c$  and a deterministic algorithm  $A$  deciding  $P$  working in time  $O(n^c)$ , where  $n$  is the size of the input.
- **NP**. Intuitively, a problem in **NP** is a problem where YES instances can be verified to be true in polynomial time when the input is completed with a "small" certificate. More precisely, a problem  $P$  is in **NP** if there exist two constants  $c, d$  and a deterministic algorithm  $A$  such that for each input  $I$  of size  $n$ ,  $P(I)$  equals true if and only there exists a certificate  $C$  of size  $O(n^c)$  such that  $A$  applied to  $I$  and  $C$  returns YES in time  $O(n^d)$ .
- **NP-hard** problem. A problem  $P$  is *NP-hard* if, for all problems  $P'$  in **NP**, there is a deterministic algorithm  $A$  working in polynomial time such that any instance of  $P'(X)$  is equivalent to  $P(A(X))$ . Thus, intuitively, a **NP-hard** problem is a problem "containing" the difficulty of all **NP**-problems.
- **NP-complete** problem. A problem  $P$  is *NP-complete* if it is both in **NP** and **NP-hard**.

First, it is obvious that  $\mathbf{P} \subseteq \mathbf{NP}$  while it is unknown whether  $\mathbf{NP} \subseteq \mathbf{P}$ . It is commonly assumed that  $\mathbf{P} \subsetneq \mathbf{NP}$ . Under this assumption, no  $\mathbf{NP}$ -complete problem can be solved in polynomial time since otherwise any other  $\mathbf{NP}$  problem would also have a polynomial time algorithm solving it.

Moreover, under the same assumption, it is reasonable to look for complexity classes that refine  $\mathbf{NP}$ . A way to do such refinements is the parameterized complexity.

**Parameterized complexity** When looking at problems, we can consider a measurement of a part of the input or output called a *parameter* that encapsulate some complexity of the considered problem. In this thesis we will restrict ourselves, as often done, to the case when the parameter is an integer. A problem together with a parameter will later be referred as a *parameterized problem*.

The idea is to express more precisely the working time of algorithms solving a parameterized problem in a function of both the size of the parameter and the size of the input. Most of the time, we look to minimize the function in the size of the input. The idea of parameterized complexity is that we expect to be able to solve the problem faster if the parameter is small (or even bounded).

For example, we define two problems that have a very intuitive parameter:

1. The VERTEX COVER problem: Given a graph  $G$  and an integer  $k$  decide whether  $G$  contains a set  $S$  of  $k$  vertices of  $G$  such that, for each edge  $\{u, v\}$  of  $G$ , we have either  $u$  or  $v$  in  $S$ . We parameterize VERTEX COVER by  $k$ , the size of the target set  $S$ .
2. The CLIQUE problem: Given a graph  $G$  and an integer  $k$  decide whether  $G$  contains a  $k$ -clique. (A  $k$ -clique is a set  $S$  of  $k$  vertices of  $G$  such that each pair of vertices of  $S$  is connected by an edge in  $G$ .) We parameterize CLIQUE by  $k$ , the size of the target  $k$ -clique.

Both problems are  $\mathbf{NP}$ -complete. Nonetheless, when parameterized, the two problems will not allow the same efficiency in the size of the graph:

1. VERTEX COVER can be decided by a deterministic algorithm that works in time  $O(2^k n)$  where  $n$  is the size of the input graph and  $k$ , the size of the target set, the parameter [Cyg+15, Section 3.4].
2. On the other hand, CLIQUE can be decided in time  $n^{O(k)}$  where  $n$  is the size of the input graph and  $k$  is the size of the target clique by verifying for each set of  $k$  vertices whether it is a clique. It is expected that we cannot do significantly better (see  $W[1]$ -hardness below).

The cases like VERTEX COVER are considered as the best we could hope for  $\mathbf{NP}$ -complete problem while the cases like CLIQUE are the ones we would like to avoid. We will then formerly define the "good" cases as *fixed parameter tractable*.

**Fixed Parameter Tractable (FPT)** A classical problem class when looking for parameterized complexity is the fixed parameter tractable problem class.

**Definition 3.5.1.** *Let  $P$  be a problem of parameter  $k$ .  $P$  is fixed parameter tractable (FPT) with respect to  $k$  if there exists a function  $f$ , a constant  $c$  and a deterministic algorithm solving  $P$  working in time  $f(k)n^c$  where  $n$  is the size of the input.*

For example, the VERTEX COVER problem parameterized by the size of the target set is FPT since there exists an algorithm working in time  $f'(k)n$  for some function  $f'$ , where  $k$  is the size of the target covering set and  $n$  the size of the input graph.

Consider an FPT problem  $P$ , all the problems  $(P_k)_{k \in \mathbb{N}}$  where  $P$  is restricted to the case where the parameter is equal to  $k$  are all problems of **P**. The converse is widely considered to be not true. For example the CLIQUE problem parameterized by  $k$  the size of the target clique is admitted to not be an FPT problem. Still, trying to find a  $k$ -clique in a graph can be done in time  $O(n^k)$ . This way, when  $k$  is fixed, we obtain a polynomial algorithm for each instance of the CLIQUE problem restricted to clique of a given size.

**$W[1]$ -hardness** When looking at **NP**-complete problems, we can try to find an FPT algorithm for some parameters as intuitive as possible for the problem. Nonetheless, it is believed that not all **NP** problems parameterized by any parameter are FPT.

The complexity classes  $W[1], \dots, W[n]$  called the  $W$ -hierarchy can be defined to extend FPT. It is commonly supposed that  $W[1]$ -hard problems are exactly problems that are not FPT. In this thesis, we will not define the classes of the  $W$ -hierarchy. Instead, we will only refer to  $W[1]$ -hard problems which will be defined as the complement of FPT problems. The reader can refer to [Cyg+15], pp. III, 13.3] for a formal definition of the  $W$ -hierarchy.

For example, the CLIQUE problem parameterized by the size of output clique is shown to be  $W[1]$ -hard. This, together with the hypothesis that  $W[1]$ -hard problems are exactly problems that are not FPT, gives that CLIQUE is not an FPT-problem. Furthermore, there is a common assumption even stronger for clique called the Exponential Time Hypothesis (the reader can see [Cyg+15], pp. III, 14] for a formal definition of it) that, in particular, assesses that there no deterministic algorithm that solve CLIQUE in time  $O(n^{o(k)})$ , with  $n$  the size of the input graph and  $k$  the size of the target clique.

Thus, finding whether an **NP**-complete problem parameterized by a given parameter is either FPT or  $W[1]$ -hard is a pretty common field of parameterized complexity.

**Slicewise polynomial (XP)** Another classic complexity class in parameterized complexity is **XP** extending the FPT class.

**Definition 3.5.2.** *Let  $P$  be a problem of parameter  $k$ .  $P$  is slicewise polynomial (XP) with respect to  $k$  if, for each  $k$ , there exists a deterministic algorithm solving  $P$  working in time polynomial in the size of the input.*

As stated before all FPT algorithms are working in polynomial time when the parameter is fixed. Thus, all **FPT** problems are also **XP** problems. The opposite is not true.

**Proposition 3.5.3.** *There exists some **XP** problems that are not fixed parameter tractable.*

### 3.5.2 Branch decomposition

A very useful tool to construct FPT algorithms on graphs is the branchwidth and the branch decomposition of a graph. The standard definition of branch decomposition uses unrooted trees while in this thesis, we will use a close variant of it where the tree is rooted.

We choose these rooted branch decompositions in order to make smoother the construction of dynamic programming algorithms (see Section [3.5.3](#)).

**Definition 3.5.4.** *Let  $G$  be a graph. A branch decomposition  $\mathcal{B}$  of  $G$  is a tree rooted at a leaf whose inner nodes are of degree 3 together with a bijective map from edges of  $G$  to non root leaves of  $\mathcal{B}$ .*

*Let  $\mathcal{B}$  be a branch decomposition and  $b$  an edge of  $\mathcal{B}$ .  $\mathcal{B} \setminus b$  is composed of two connected components  $\mathcal{B}_1$  containing the root and  $\mathcal{B}_2$ . Then, it induces a partition of edges of  $G$  (if  $b$  is the arc incident to the root, then one part of the partition is empty) in two subgraphs  $G_1$  and  $G_2$  that are the graphs induced by the edges labeling the leaves of respectively  $\mathcal{B}_1$  and  $\mathcal{B}_2$ .  $G_1$  and  $G_2$  are respectively called the upper part and the lower part of the bipartition of  $G$  associated with  $b$ . The set  $S$  of vertices of  $G$  incident to both at least one edge of  $G_1$  and at least one of  $G_2$  is the middle set of  $b$ . The width of  $\mathcal{B}$  is the maximum of the cardinality of the middle set of an edge of  $\mathcal{B}$ .*

*Eventually, the branchwidth of  $G$  is the minimum on all branch decompositions  $\mathcal{B}$  of  $G$  of the width of  $\mathcal{B}$ .*

See Figure [3.6](#) for an example of branch decomposition.

The usual definition of a branch decomposition is identical, except that the tree is unrooted, and thus the leaves are in bijection with the edges of  $G$ . The difference is cosmetic: From any usual branch decomposition, one can trivially obtain a rooted branch decomposition of the same width, by subdividing an arbitrary arc with a new node  $\nu$  and then connecting  $\nu$  to a new leaf node  $\rho$ , which will serve as the root; the converse operation obviously transforms any rooted branch decomposition into a usual branch decomposition. Since each usual branch decomposition corresponds to a rooted branch decomposition, and both have the same width, we henceforth only work with rooted branch decompositions.

**Connections with the treewidth** Related to branch decomposition, and perhaps more well-known, is the notion of tree decomposition. We will not use (nor define) tree decomposition and treewidth in this thesis. Let us, however, remark that the treewidth is similar to branchwidth in two ways. First, proofs tactics for one can often be adapted

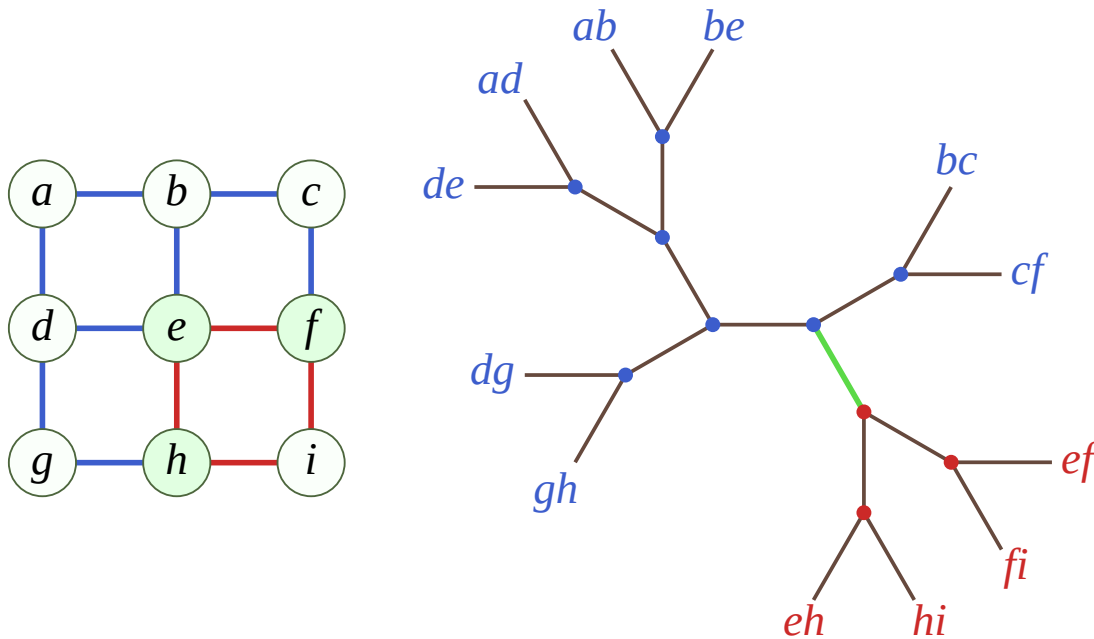


Figure 3.6: A branch decomposition of a grid graph showing a partition of width 3 corresponding to an edge. (By: David Eppstein)

to the other. Second, for all graphs  $G$ , its treewidth  $tw$  and its branchwidth  $bw$  are comparable as follows [RS91]:

$$bw \geq 2 \Rightarrow bw \leq tw + 1 \leq \frac{3}{2}bw$$

In this thesis, branch decomposition is preferred to tree decomposition for one main reason: dynamic programming strategies are often easier to define with branch decomposition.

**Complexity of computing a branch decomposition** We define the BW problem: given a graph  $G$  and an integer  $b$ , decide whether there exists a branch decomposition of width at most  $w$  of  $G$ .

BW has been shown to be both an NP-complete problem and a FPT problem when parameterized by  $b$  the target width [BT97]. Moreover, the FPT algorithm deciding BW returns a branch decomposition of minimal width.

### 3.5.3 Dynamic programming

One key use of the branchwidth is a proof strategy to construct FPT algorithms: dynamic programming.

**Principle of dynamic programming** Let  $P$  be a problem applied to a graph  $G$  containing  $n$  edges. Let  $\mathcal{B}$  be a branch decomposition of  $G$  of width  $w$ , the branchwidth of  $G$ , rooted on a leaf  $r$ .

The idea is to first find some subproblems expressed on the middle set of edges of  $\mathcal{B}$  such that:

1. Solutions of the subproblems of an edge  $e$  incident to non-root leaves can be constructed in time  $f(w)$  independent of  $G$ .
2. There exists a function  $g$  and a constant  $c_1$  independent of  $G$  such that solutions of the subproblems of an edge  $e$  not incident to a non-root leaf can be constructed from the solutions of the subproblems of other edges incident to the node descending from it in time  $O(g(w)n^{c_1})$ .
3. A solution of the problem can be deduced from the solutions of subproblems on the edge incident to the root in constant time  $h(w)$  independent of  $G$ .

With these subproblems, we can define the dynamic programming strategy:

- First, thanks to 1, we can, by brute force, construct all solutions of subproblems of edges incident to non-root leaves.
- Then, thanks to 2, we construct solutions of the subproblems for other edges from the ones of the edges below it by bottom-up induction.
- Eventually, thanks to 3, we deduce a solution for the general problem from the solutions of the edge incident to the root.

Moreover,  $\mathcal{B}$  contains  $n - 1$  edges incident to non-leaf edges and  $n$  edges incident to leaves. Thus, the dynamic programming algorithm is an FPT algorithm since it works in time  $O(g(w)n^{c_1+1} + f(w)n + h(w))$ .

**INDEPENDENT SET example** We will give an example of *dynamic programming* used to construct an FPT algorithm solving INDEPENDENT SET. A set  $S$  of vertices of a graph  $G$  is an *independent set of  $G$*  if there is no edge between two vertices of  $S$  in  $G$ . We define INDEPENDENT SET: Given a graph  $G$  and an integer  $k_0$  decide whether there exists an independent set of  $G$  of cardinality at least  $k_0$ . We will parameterize INDEPENDENT SET by  $w$  the branchwidth of  $G$ .

We want to use dynamic programming to construct an FPT algorithm solving INDEPENDENT SET.

Let  $(G, k_0)$  be an instance of INDEPENDENT SET.

We first use a FPT algorithm to construct a branch decomposition  $\mathcal{B}$  of  $G$  of width  $w$ , the branchwidth of  $G$  and root it at some leaf  $r$ . For each edge  $b$  of  $\mathcal{B}$ , let  $V(b)$ ,  $G_b^-$  and  $G_b^+$  be respectively the middle set of  $b$ , the lower and the upper part of the bipartition of  $G$  associated with  $b$ .

Then, the objects we will consider are, for each edge  $b$  of  $\mathcal{B}$  and each set  $S \subseteq V(b)$ , the integer  $m(S, b)$  which will be the maximal size of an independent set  $S'$  of  $G_b^-$  such

that  $S' \cap V(b) = S$ . If there is no independent set containing  $S$ , then we will put  $m(S, b) = -\infty$ .

1. First, consider an edge  $b$  incident to leaf  $l$  other than the root and  $\{u, v\}$  the edge associated with  $l$ . Then,  $G_b^-$  is only the edge  $\{u, v\}$  and  $V(b) = \{u, v\}$ . Thus there are four sets included in  $V(b)$  with:  $m(\emptyset, b) = 0$ ,  $m(\{u\}, b) = m(\{v\}, b) = 1$  and  $m(\{u, v\}, b) = -\infty$ .
2. Then, we construct the core dynamic programming. Consider an edge  $b$  incident to an internal node  $t$  incident to  $b_1$  and  $b_2$ . We suppose that we have already constructed all  $m(S, b_1)$  and all  $m(S, b_2)$ . Then, let  $S$  be a subset of  $V(b)$ . We consider all sets  $S_1 \subseteq V(b_1)$  and all sets  $S_2 \subseteq V(b_2)$  such that  $S_1 \cap V(b) = S \cap V(b_1)$ ,  $S_2 \cap V(b) = S \cap V(b_2)$  and  $S_1 \cap V(b_2) = S_2 \cap V(b_1)$  and put the maximum among all  $S_1, S_2$  of  $m(S_1, b_1) + m(S_2, b_2) - |S_1 \cap S_2|$  as  $m(S, b)$ .  
Since both  $V(b_1)$  and  $V(b_2)$  are of cardinality at most  $w$ , there are at most  $2^{w+1}$  different ordered pairs of sets  $(S_1, S_2)$ . Thus, by testing all possible couples, all  $m(S, b)$  can be constructed in time  $O(2^w)$ . Moreover, an easy induction (see [Cyg+15], Section 7.3.1] for details) shows that the value constructed for  $m(S, b)$  is correct.
3. Eventually, we have constructed all  $m(S, b_r)$  where  $b_r$  is the edge incident to the root. Then,  $G_{b_r}^- = G$ . Thus, the maximum of  $m(S, b_r)$  among all  $S$  is the maximal weight of an independent set.

Since  $\mathcal{B}$  contains  $2n - 1$  edges where  $n$  is the number of edges of  $G$ , the dynamic programming algorithm works in time  $O(2^{wn})$ .

### 3.5.4 Irrelevant vertex method

Consider a graph problem  $P$  known to be FPT in the branchwidth that we want to parameterize by another parameter  $k$ .

If we could bound the branchwidth by  $k$ , then we would immediately have a FPT algorithm in  $k$ . Nonetheless, most of the time, it is impossible to limit the branchwidth by  $k$ .

Thus, instead, we replace an instance by an equivalent one with a small branch decomposition to then use the algorithm FPT in the branchwidth. A way to do so is the *irrelevant vertex method*.

**Excluded Grid Theorem** The first step of any irrelevant vertex method is to apply a version of the Excluded Grid Theorem.

The Excluded Grid Theorem is a vastly studied meta theorem that ensures that any graph either have small branchwidth or contains a big grid as a minor. The Excluded Grid Theorem states:

**Theorem 3.5.5.** [RS86a] *There exists a function  $f$  such that for any integer  $g > 1$ , any graph of branchwidth at least  $f(g)$  contains a  $g \times g$  grid of size  $g$  as a minor.*



The two main goals in the study of the Excluded Grid Theorem are either to find the smallest theoretical  $f$  possible such that the property holds or to construct an algorithm that given a graph constructs either a branch decomposition of width at most  $f(g)$  or a  $g \times g$  grid minor for the smallest  $f$  possible. For example, [Chu15] showed that it is possible to find a function  $f = O(g^{37})$  that applies to all graphs and gave a probabilistic algorithm for the same  $f$  that returns either a branch decomposition of size  $f(g)$  or a grid minor of size  $g$  working in expected polynomial time.

**Principle of the irrelevant vertex method** First, we use an FPT algorithm of Excluded Grid Theorem for some function  $f$ .

We consider an integer  $g(k)$  large enough. Let  $G$  be an input graph.

We use the Excluded Grid Theorem to find either a branch decomposition of width at most  $f(g(k))$  or a grid of size  $g(k)$ . Then, in the second case, we find a vertex  $v$  in the grid such that the problem is equivalent when replacing  $G$  by  $G \setminus v$ . Then,  $v$  is said to be an *irrelevant vertex*.

When obtaining an irrelevant vertex, we remove it and repeat the process until we obtain an instance with a branch decomposition of size at most  $f(g(k))$ .

This altogether gives an FPT algorithm to obtain an equivalent input of small branch-width.

**Example: PLANAR ODD CYCLE PACKING** We will give an example of *irrelevant vertex method* on PLANAR ODD CYCLE PACKING. First, we define PLANAR ODD CYCLE PACKING: given a planar graph  $G$  and an integer  $k$ , decide whether there exists in  $G$  a set of  $k$  cycles of odd length not sharing any vertex. We will parameterize PLANAR ODD CYCLE PACKING with  $k$ , the number of desired cycles.

Let  $A$  be an algorithm that, for some function  $f$ , finds in a graph either a branch decomposition of width at most  $f(k)$  or a grid minor of size  $2\sqrt{k}(2k+1)$ .

Consider an instance  $(G, k)$ . We want to replace it by an equivalent instance  $(G', k)$  with a branch decomposition  $B'$  of  $G'$  of width at most  $f(k)$ . To obtain this, we apply inductively the following process which is an *irrelevant vertex method*:

- First, we use  $A$ .
- If it returns a branch decomposition  $B$  of  $G$  of width at most  $f(k)$ , we can choose  $G' = G$  and  $B' = B$ .
- Otherwise, we have a wall  $W \subseteq G$  of size at least  $2\sqrt{k}(2k+1)$ . Then, we can easily find in  $W$ ,  $k$  subwalls  $W_1, \dots, W_k$  of size  $2(k+1)$  that are disjoint from one another. Let  $C_1, \dots, C_k$  be the outer cycles of  $W_1, \dots, W_k$ . If all are of odd length, we have  $k$  cycles of  $G$  of odd length not sharing any vertex. Thus, the answer is YES.
- Otherwise, let  $W_0$  be one of the subwalls with an outer cycle of even length. Then, we can find  $2k$  disjoint cycles  $C'_1, \dots, C'_{2k}$  inside  $W_0$  nested in one another such that no two vertices of two distinct cycles are adjacent together with a vertex  $v_0$

nested inside all not incident to any cycle. If at least  $k + 1$  cycles are of odd length, then the answer is YES.

- Otherwise,  $v_0$  is irrelevant, meaning that  $G$  contains  $k$  disjoint odd cycles if and only if  $G \setminus v_0$  does too. We refer to [Gol+09, Lemma 4] for a proof of it.

Then, we repetitively apply the previous method until we obtain either the YES answer or an instance where the branchwidth of the graph is at most  $f(k)$ .



# Topological embedding problems

Often, simplicial complexes are used as a combinatorial representation of a topological space. Thus, the following problem is a widely studied one: Given two simplicial complexes  $S$  and  $T$ , determine whether there exists a topological embedding of  $S$  on  $T$ .

In the following chapters of the thesis we will study a restriction of the previous problem: Given a graph  $G$  and a 2-complex  $\mathcal{C}$ , determine whether there exists a topological embedding of  $G$  on  $\mathcal{C}$

In this chapter, we will present a quick overview of results known for some embedding problems on simplicial complexes. This includes embeddings into the  $d$ -dimensional ball, since the  $d$ -dimensional ball is homeomorphic to the  $d$ -dimensional simplicial complex consisting of a single simplex of dimension  $d$  and all its faces. In order to keep this survey short, we first exclusively focus on topological embedding problems in which geometry plays no role. We then describe problems that have a similar flavor, but which do not fall exactly into the class of topological embedding problems.

## 4.1 Topological embeddings on simplicial complexes

### 4.1.1 Embedding simplicial complexes into the $d$ -ball

One of the main cases where topological embedding problems have been studied is the case where the target space is the ball of dimension  $d$ .

Formally, for any fixed integers  $k$  and  $d$ , let us define the problem  $EMBED_{k \rightarrow d}$ : given a  $k$ -dimensional complex  $\mathcal{C}$ , decide whether  $\mathcal{C}$  can be embedded in the  $d$ -ball.

First, there are two cases where  $EMBED_{k \rightarrow d}$  is trivial:

- $d < k$ : The  $(d + 1)$  simplex cannot be embedded on the  $d$  ball and, thus no  $k$ -dimensional complex can be embedded on  $\mathbb{R}^d$ .
- $d \geq 2k + 1$ : All  $k$ -dimensional complexes are embeddable on  $\mathbb{R}^d$ . In fact, building an embedding is easy. Let  $\mathcal{C}$  be a simplicial complex of dimension  $k$ . First, we can choose any embedding  $\Pi$  of the vertices of  $\mathcal{C}$  in general position (for every

$i$ -dimensional affine space contains at most  $i + 1$  vertices). Then, extending  $\Pi$  by mapping each simplex of  $\mathcal{C}$  to the convex hull of its vertices gives an embedding of  $\mathcal{C}$ .

This case applied to  $d = 3$  and  $k = 1$  gives that all 1-dimensional complexes, i.e. graphs, are embeddable on any 3-dimensional complex.

Otherwise, when  $d \geq k \geq \frac{d-1}{2}$ , some but not all  $k$ -dimensional complexes can be embedded in  $\mathbb{R}^d$ . There are three different cases, each of them is studied with very different techniques:

- $k \leq d < 4$ : This is the low dimension case. When applied to  $k = 1$  and  $d = 2$ ,  $EMBED_{1 \rightarrow 2}$  is graph planarity testing. J.Hopcroft and R.Tarjan [HT74] gives an algorithm deciding  $EMBED_{1 \rightarrow 2}$  in linear time. Moreover,  $EMBED_{2 \rightarrow 2}$  is known to be decidable in polynomial time [MTW09, Appendix A]. Furthermore, both problems  $EMBED_{2 \rightarrow 3}$  and  $EMBED_{3 \rightarrow 3}$  are known to be decidable [Mat+14].
- $\frac{d}{2} \leq k < \frac{2d-2}{3}$ : This is known as the meta-stable range (which only exists when  $d \geq 4$ ). The meta-stable range is the "easy" non-trivial case as for all couples  $k, d$  of the meta-stable range,  $EMBED_{k \rightarrow d}$  is decidable in polynomial time [CKV17].
- $d \geq 4$  and  $\frac{2d-2}{3} \leq k \leq d$ : Then, the problem is known to be **NP**-hard.  $EMBED_{k \rightarrow d}$  is even known to be undecidable when  $d \geq 5$  and  $d \geq k \geq d - 1$  [MTW09].

Finally, a small remark: The aforementioned papers deal with embeddability on  $\mathbb{R}^d$ , but the algorithms immediately extends to the embeddability in the  $d$ -ball too.

### 4.1.2 Embedding graphs on surfaces

Another widely studied case is when the target space is a surface (and thus, without loss of generality, the source space is a 2-complex). In this section, we restrict ourselves to graphs while the general case of embedding of 2-complexes on surfaces is described the next section.

Embedding graphs on surfaces have been quite largely studied. Deciding if a graph is embeddable on an arbitrary surface is **NP**-complete [Tho89]. So, in the spirit of parameterized complexity, one can consider the genus of the surface as a parameter and hope to find efficient algorithms when the genus is small or fixed. This is indeed the case. There are three main approaches to attack this problem: the graph minor approach, the approach by Mohar [Moh99] and the one by Kawarabayashi, Mohar and Reed [KMR08], which we detail below.

On the one hand, the graph minor approach induces an algorithm working in time quadratic in the size of the graph to embed it on a fixed surface.

On the other hand, the two other algorithms decide whether a graph  $G$  is embeddable on a surface  $\mathcal{S}$  are working in time linear in the size of  $G$  and at least exponential in the genus of  $\mathcal{S}$ . These algorithms are thus FPT in the genus of  $\mathcal{S}$  while being very efficient when  $\mathcal{S}$  is fixed. Both use the fact that any graph embeddable on a surface is

embeddable on any surface with greater genus and same orientability. Thus, both papers are in fact giving the smallest genus of a surface with the wanted orientability on which the input graph can be embedded.

**The graph minor approach** A key property of the class of graphs embeddable on a fixed surface is that it is minor-closed: Given a graph  $G$  embeddable on a surface  $\mathcal{S}$ , all minors of  $G$  are embeddable on  $\mathcal{S}$ . Due to this, the graph minor theorem by Robertson and Seymour [RS04] stipulates that, for each surface  $\mathcal{S}$ , there is a finite set of graphs  $\mathcal{G}_{\mathcal{S}}$  such that a graph  $G$  is embeddable on  $\mathcal{S}$  if and only if  $G$  does not admit any graph of  $\mathcal{G}_{\mathcal{S}}$  as a minor. Thus, the graphs of  $\mathcal{G}_{\mathcal{S}}$  are called the *forbidden minors* of  $\mathcal{S}$ . Since deciding whether a fixed graph  $M$  is a minor of an input graph  $G$  takes quadratic time in the size of  $G$  [KKR12], this leads to a quadratic-time algorithm to decide embeddability of a graph on a fixed surface.

**Overview of the algorithm of [Moh99]** For simplicity, we will present an overview of [Moh99] only on the orientable case: The non-orientable one is handled similarly but is a bit more technical.

First, a *branch* of a graph  $G$  is either (a) a maximal connected subgraph of  $G$  whose vertices are all of degree one or two, or (b) a vertex of degree 3 or more. Then, for each triple of integers  $g, k, n$ , we define the cellular *embedding extension problem (EEP)* on graphs with few branches EEP-CELL( $g, k, n$ ):

- Input: a graph  $G$  of size at most  $n$ , a subgraph  $H$  of  $G$  composed of at most  $k$  branches such that each connected component of  $G \setminus H$  intersects several branches of  $H$ , and a cellular embedding  $\Pi_H$  of  $H$  on a surface  $\mathcal{S}$  of genus  $g$ .
- Output:
  - Return an embedding  $\Pi$  of  $G$  on  $\mathcal{S}$  such that  $\Pi$  restricted to  $H$  is equal to  $\Pi_H$  when there exists such a  $\Pi$ .
  - Return a subgraph  $H'$  of  $G$  containing  $H$  with a number of branches bounded by a function of  $k$  and  $g$  such that there exists no embedding of  $H'$  extending  $\Pi_H$  otherwise.

The following property of EEP-CELL is a key ingredient of the paper that we will reuse in Chapter 6.

**Proposition 4.1.1.** *There is a function  $f : \mathbb{N}^2 \rightarrow \mathbb{N}$  such that every instance of EEP-CELL( $g, k, n$ ) can be solved in time  $f(k, g) \cdot O(n)$*

*Proof.* The ideas are presented in [Moh99], p.23-24]

The algorithm first reduces the problem to an equivalent instance  $(G', H', \Pi', \mathcal{S})$  where  $H'$  still satisfies that each connected component of  $G' \setminus H'$  intersects several branches of  $H'$  while also satisfying that the vertices of  $H'$  contains all vertices of  $H$  of degree different from two (all these are also vertices in  $H'$  of degree different from two) plus a

constant number of vertices of degree two in  $H'$ , where this constant number is bounded from above in terms of the genus of  $\mathcal{S}$ . For this purpose, [Moh99] relies on another paper [JMM97]. After achieving this property, the paper reduces the EEP to a constant number (where the constant depends on  $g$ ) of “simple” extension problems [Moh99, Section 4]. The EEPs are then solved in [Moh99, Theorem 5.4] which relies on several other papers [JMM07; JM05; Moh06].  $\square$

The general algorithm starts by constructing a subgraph  $H_0$  with a bounded number of branches such that each connected component of  $G \setminus H_0$  intersects several branches of  $H_0$ .

Iteratively, it finds the minimal genus on which  $H_0$  is embeddable by trying one by one surfaces of increasing genus. For each cellular orientable embedding  $\Pi_{H_0}$  of minimal genus of  $H_0$ , it solves EEP-CELL  $(G, H_0, \Pi_{H_0})$ . If at least one  $\Pi_{H_0}$  induces a positive answer, then it is an embedding of  $G$ . Otherwise, let  $H_1$  be the union of all subgraphs returned by the different instances of EEP-CELL considered. It replaces  $H_0$  by  $H_1$ , and repeats the previous operation until it obtain an embedding of minimal genus of  $G$ .

The algorithm works in time at least doubly exponential on the genus of the surface while linear in the size of  $G$ .

**Overview of the algorithm of [KMR08].** The approach of [KMR08] is very different from [Moh99] but the strategy for the non-orientable case is also pretty much the same as the one for the orientable one simply more technical. So, as in Paragraph 4.1.2, we will also present an overview of [KMR08] only on the orientable case.

The extended abstract [KMR08] provide a completely different strategy that does not use embedding extension problem and works in time simply exponential in the genus of the surface. This algorithm uses an irrelevant vertex method to limit the treewidth and then uses a dynamic programming algorithm to solve the problem when the treewidth is bounded.

The idea is to construct first a certain sequence of graphs  $G_0, G_1, \dots, G_b$  of decreasing size where  $G_0$  is the input graph and  $G_b$  is of bounded size and thus easy to embed on a surface with the minimal genus, where  $b$  depends only on the genus targeted.  $G_{i+1}$  is obtained from  $G_i$  by either removing a large independent set or contracting a large matching. Both transformations induce a reverse transformation giving  $G_i$  from  $G_{i+1}$ .

Then, this sequence is used to construct backwards another sequence of graphs  $H_b, \dots, H_0$  of bounded treewidth such that:

1.  $H_b = G_b$
2.  $H_i$  is a subgraph of  $G_i$  with the same genus as  $G_i$  and bounded treewidth.
3. An embedding of minimal genus of  $H_i$  can be constructed from one of  $H_{i+1}$ .
4. An embedding of  $G_i$  can be deduced from one of  $H_i$ .

In order to construct  $H_i$  from  $H_{i+1}$ , we define first an intermediate graph  $\tilde{H}_i$ .  $\tilde{H}_i$  is the subgraph obtained by restricting the transformation giving  $G_i$  from  $G_{i+1}$  to  $H_{i+1}$ . Then, there are two cases:

- If  $\tilde{H}_i$  is of genus at least  $g + 1$ : it finds a subgraph of  $G_i$  not embeddable on the surface of genus  $g$ .
- If  $\tilde{H}_i$  is of genus at most  $g$ : It constructs an irrelevant vertex method that finds and removes irrelevant vertices (See Section 3.5.4) of  $\tilde{H}_i$  for the embeddability of  $G_i$  until  $\tilde{H}_i$  is of bounded treewidth. Then,  $H_i$  is the final  $\tilde{H}_i$ .

For the third point, it gives a method to construct a cellular embedding of minimal genus of  $H_i$  from one of  $H_{i+1}$  and a branch decomposition of  $H_i$ .

Eventually, it shows that  $G_i$  have the same genus as  $H_i$  and how an embedding of  $G_i$  can be deduced from an embedding of  $H_i$ .

Thus, it returns an embedding of  $G = G_0$  of minimal genus if this one is smaller than  $g$ .

Unfortunately, [KMR08] is an extended abstract which never materialized into a full version. While the general strategy is clear, the details are elusive and hard to check [KP19, Footnotes]. Our algorithm in Chapter 7 uses the same general strategy for embedding graphs on 2-complexes, and in particular provides a complete proof for embedding graphs of bounded treewidth (or branchwidth) on surfaces of bounded genus in linear time.

### 4.1.3 Embedding 2-complexes on surfaces

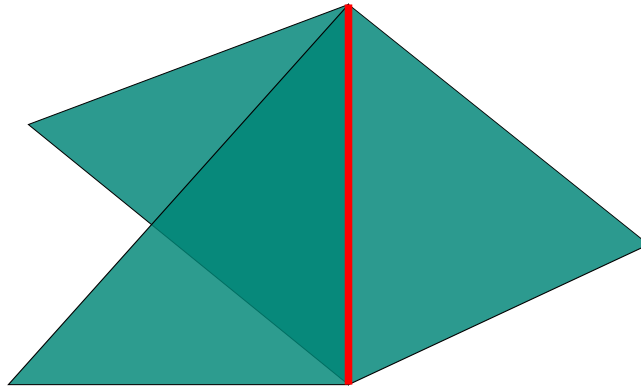


Figure 4.1: An example of 3-book with the spine edge in red.

After looking at embedding 1-complexes on surfaces, it is natural to look for embedding of 2-complexes on surfaces. To do so B.Mohar [Moh97] uses the embeddability of the 1-skeleton of refinements of the 2-complex we try to embed.



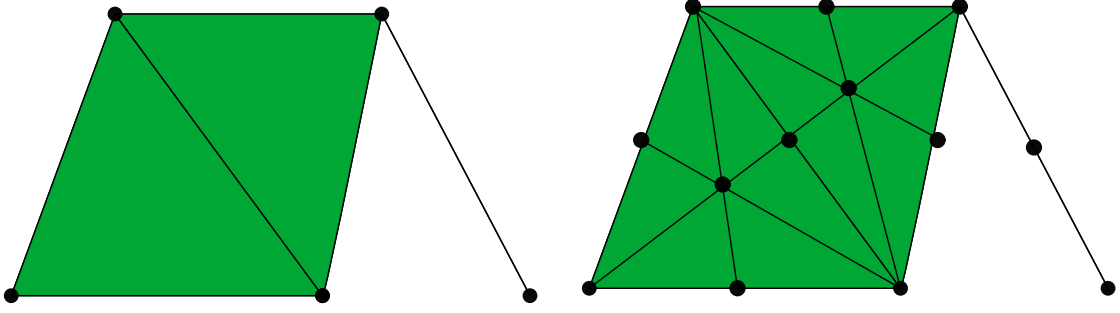


Figure 4.2: A 2-complex on the left and its barycentric subdivision on the right

First, not all 2-dimensional complexes are embeddable on some surface. For example, the simplicial complex formed by three triangles sharing an edge known as the 3-book (See Figure 4.1), is not embeddable on any surface.

On a surface, each point admits a neighborhood homeomorphic to a disk. So, as an obviously necessary condition, any 2-complex  $\mathcal{C}$  embeddable on some surface satisfies that a neighborhood of each vertex of  $\mathcal{C}$  is embeddable on a disk. Gross and Rosen [GR81] prove that the condition is sufficient.

In the rest of this section, we will only consider 2-complexes embeddable on some surface. When considering such a 2-complex  $\mathcal{C}$ , we can look for the minimal genus of a surface on which  $\mathcal{C}$  is embeddable. As for graphs, we can define:

- The orientable genus of  $\mathcal{C}$  by considering embeddings on orientable surfaces only;
- The non-orientable genus of  $\mathcal{C}$  by considering embeddings on non-orientable surfaces only;
- The Euler genus of  $\mathcal{C}$  by considering embeddings on both orientable and non-orientable surfaces.

In the rest of the section we will use the barycentric subdivision of a 2-complex which is the 2-complex where each edge is subdivided and each triangle is subdivided in 6 triangles See Figure 4.2

Obviously, if a 2-complex  $\mathcal{C}$  is embeddable on a surface  $\mathcal{S}$ , then all barycentric subdivisions of  $\mathcal{C}$  and in particular their 1-skeletons (the graph formed by simplices of dimension one or less of the subdivision) are also embeddable on  $\mathcal{S}$ . This naturally raises the question to know if the converse holds: if the 1-skeleton of all subdivisions of a 2-complex  $\mathcal{C}$  are embeddable on a surface  $\mathcal{S}$  then is  $\mathcal{C}$  embeddable on  $\mathcal{S}$ ? In this spirit, Mohar [Moh97] proved the following results:

- Surprisingly, when looking for either the non-orientable or the Euler genus of a 2-complex  $\mathcal{C}$  we can simply consider the 1-skeleton of the fifth subdivision. Thus, the minimal non-orientable and the minimal Euler genus of a surface on which  $\mathcal{C}$  is embeddable are the same as the one of the 1-skeleton of the fifth barycentric subdivision of  $\mathcal{C}$ .

- For the orientable genus, subdividing a constant number of times does not work; it is necessary and sufficient to subdivide  $\mathcal{C}$   $\Theta(\log(g))$  times. More precisely, for each  $g$ , Mohar [Moh97] gives a 2-complex  $\mathcal{C}_0$  not embeddable on the orientable surface of genus  $g$  such that the 1-skeleton of the  $\lceil \log_2(g) \rceil$ th subdivision of  $\mathcal{C}_0$  is embeddable on that surface. Nonetheless, let  $\mathcal{C}$  be a 2-dimensional complex embeddable on some orientable surface. Then,  $\mathcal{C}$  is embeddable on the orientable surface of genus  $g$  if and only if the one skeleton of the  $(1 + \lceil \log_2(g+2) \rceil)$ th barycentric subdivision of  $\mathcal{C}$  is too.

The first point, combined with an FPT algorithm for embeddability of graphs on surfaces [KMR08; Moh99], yields an FPT algorithm to decide given a 2-complex  $\mathcal{C}$  and a non-orientable surface  $\mathcal{S}$  whether  $\mathcal{C}$  is embeddable on  $\mathcal{S}$ .

Nevertheless, there is no known FPT algorithm deciding whether a 2-complex is embeddable on an orientable surface.

## 4.2 Related problems

### 4.2.1 Problems in the plane

A lot of problems where the target space is the plane that are close to topological embedding have been studied.

**Crossing number** Instead of considering embeddings, we can consider general position drawings. A general position drawing of a graph  $G$  on the plane is a mapping from  $G$  to the plane where the images of edges are allowed to intersect except that:

- no more than two edges can intersect in any point except at vertices,
- edges do not intersect vertices except at their endpoints,
- the set of points where two edges intersect is finite.

Given a graph  $G$ , its *crossing number* is the minimal number of crossings (points where two edges intersect out of their endpoints) of a general position drawing of  $G$  on the plane. This naturally leads to the following algorithmic problem: given a graph  $G$  of size  $n$  and an integer  $k$ , decide whether  $G$  have crossing number at most  $k$ . Kawarabayashi and Reed [KR07] gave a FPT algorithm solving this problem working in time linear in the size of  $G$  when  $k$  is fixed.

**Planarity number of a graph** As studied by Bienstock and Monma [BM88], we can also look for the planarity number of a graph  $G$ . First, we define the planarity number of a planar embedding. Let  $\Pi$  be a planar embedding of a connected graph  $G$ ; the planarity number of  $\Pi$  is the minimal cardinality of a set  $\mathcal{F}$  of faces of  $\Pi$  such that all vertices of  $G$  appear in the facial walk of at least one of the faces of  $\mathcal{F}$ . The planarity number of  $G$  is the minimal planarity number among all planar embeddings of  $G$ . Bienstock and

Monma [BM88] gave an algorithm taking an integer  $k$  and a graph  $G$  deciding whether the planarity number of  $G$  is at most  $k$  and, if it is the case, constructing a planar embedding  $\Pi$  and a set of faces  $F$  of  $\Pi$  realizing the planarity number of  $G$ . This algorithm is FPT in the planarity number of  $G$ .

#### 4.2.2 Removing vertices to lower the genus: a generalization of embedding graphs on surfaces

Using a linear time algorithm embedding graphs on surfaces [Moh99; KMR08] as a black box, Kociumaka and Pilipczuk [KP17] constructs a fixed parameter tractable algorithm solving a problem a bit more general. Consider a graph  $G = (V, E)$ , a positive integer  $k$  and a surface  $\mathcal{S}$ . Then, they gave an algorithm that:

- decides whether there exists a set  $V_0$  of at most  $k$  vertices of  $G$  such that the graph  $G - V_0$  is embeddable on  $\mathcal{S}$ ;
- when the answer is positive, returns both such a set  $V_0$  and an embedding of  $G \setminus V_0$ .

To achieve this, the article presents an irrelevant vertex method to find either:

- A small tree decomposition of  $G$ .
- A vertex  $v$  such that all solutions for  $G$ ,  $k$  and  $\mathcal{S}$  are obtained by adding  $v$  to one for  $G - v$ ,  $k - 1$  and  $\mathcal{S}$ . Thus, we replace  $G$  by  $G - v$  and  $k$  by  $k - 1$ .

Then, finding a solution for  $G$ ,  $k$  and  $\mathcal{S}$  is equivalent to finding one for  $G - v$ ,  $k - 1$  and  $\mathcal{S}$  by simply adding/removing  $v$  from  $G$  and the set  $V_0$  of a solution. Thus, it replaces  $G$  by  $G - v$  and  $k$  by  $k - 1$ .

- An *irrelevant vertex*  $v$  for the considered problem. Then,  $G$  can be replaced by  $G - v$  without loss of generality.

Then, it applies this method until it obtains either  $k < 0$  or a small tree decomposition. In the first case, the answer is negative. In the second case, it gives a dynamic programming algorithm solving the problem thanks to the tree decomposition.

Our algorithm in Chapter 7 will use the same general strategy of first giving an algorithm that either finds a small branch decomposition, a subgraph not embeddable on the target 2-complex, or an irrelevant vertex, and then giving an algorithm of dynamic programming using the small branch decomposition.

#### 4.2.3 Problems on 2-complexes

Although the embedding problem on 2-complexes itself had not been studied prior to our work, related problems were considered.

**Finding a sphere in a 2-complex** Consider the following problem: given a 2-dimensional complex  $\mathcal{C}$  and an integer  $k$ , decide whether there exists a set of at most  $k$  triangles  $S$  of  $\mathcal{C}$  such that the 2-complex formed by the triangles of  $S$  together with their faces is homeomorphic to a sphere. Burton, Cabello, Kratsch and Petterson [Bur+19] showed that this problem is  $W[1]$  hard. This would imply that there is no FPT algorithm to solve the problem (assuming  $FPT \subsetneq W[1]$ ).

**Equivalence between 2-complex homeomorphism and graph isomorphism** ÓDúnlainig, Watt and Wilkins [DWW00] gave a two-way polynomial time reduction between the two following problems:

- Graph isomorphism: given two simple graphs  $G_1$  and  $G_2$ , decide whether there exists a bijective map  $f$  between the vertices of  $G_1$  and the vertices of  $G_2$  such that there is an edge between two vertices  $u, v$  of  $G_1$  if and only if there is an edge between  $f(u)$  and  $f(v)$ .
- 2-complex homeomorphism: given two 2-dimensional complexes  $\mathcal{C}_1$  and  $\mathcal{C}_2$ , decide whether there exists a homeomorphism between  $\mathcal{C}_1$  and  $\mathcal{C}_2$ .

Moreover, graph isomorphism is a well-studied problem in **NP** that is neither known to be **NP**-complete nor solvable in deterministic polynomial time (like the integer prime factorization problem or the unknotting problem). Thus, the same holds for homeomorphism of 2-complexes.

**Book embeddings** There is one particular kind of embedding of a graph on some 2-complexes that has been studied: book embeddings. A *book with  $k$  pages* is a 2-complex formed of  $k$  triangles attached together along a single common edge called the *spine* of the book (See Figure 4.1 for an example of 3-book). A  *$k$  page book embedding* of a graph  $G$  is an embedding of  $G$  on the book with  $k$  pages such that all vertices are mapped on the spine and each edge is mapped to a single triangle. The *pagenumber* of a graph  $G$  is the minimal  $k$  such that there exists a  $k$  page book-embedding of  $G$ . It is natural to search to bound the pagenumber of a graph with its other numeric characteristics. For example, the pagenumber of a graph of treewidth  $k$  is at most  $k + 1$  [GH01], the pagenumber of a planar graph is most 4 and the pagenumber of a graph of genus  $g$  is dominated by  $\sqrt{g}$  [Mal94]. There is an abundant literature on the subject of book embedding. We refer to Dujmovic and Wood [DW04, Tab1] for an overview of results known on book embedding.

Nonetheless, later we will look for a general embedding of graphs on 2-complexes that differ from book embeddings in two main ways:

1. Vertices will be allowed to be mapped on any part of the 2-complex and not only the spine
2. The image of edges will be allowed to intersect the inside of any number of triangles.

In conclusion, we have seen that many embedding problems involving simplicial complexes have been studied. From the next chapter and onwards, we will focus on the algorithmic problem studied in this thesis, the embeddability of graphs on 2-dimensional simplicial complexes. Specifically, in the next chapter, we will explain why this problem is important in this context.

# Embeddings on 2-complexes: motivations and basic tools

We formally define the central problem of the thesis, `EMBED` : given a graph  $G$  and a 2-complex  $\mathcal{C}$ , decide whether there exists a topological embedding of  $G$  on  $\mathcal{C}$ . See Figure 1.4 for an example of an embedding on a 2-complex.

In the thesis, we will give two algorithms to solve this problem:

- One based on a topological reduction to instances of embedding extension problem on surfaces (see Section 4.1.1) which can be solved by the algorithm of Mohar [Moh99].
- A FPT algorithm using an irrelevant vertex method to obtain a small branch decomposition and a dynamic programming algorithm solving `EMBED` on graph of small branchwidth.

## 5.1 Motivations

As we shall see, `EMBED` is a natural problem to study. First, it is a very general problem, since, on the one hand, the class of simplicial complexes of dimension at most two is in a sense the most general class of topological spaces into which the graph embedding problem makes sense and, on the other hand, `EMBED` encompasses several problems studied earlier in an unified way.

Moreover, the strategies to solve the embedding problem for graphs on surfaces do not extend easily to the case of 2-complexes.

### 5.1.1 Generality of the problem

**Maximality of dimension 2 for the graph embedding problem.** Simplicial complexes allow the representation of essentially any reasonable topological space. Thus, at first sight one could consider that the most general graph embedding problem is that in

which the target space is an arbitrary simplicial complex. However, all graphs can be embedded into the 3-simplex (the solid tetrahedron, see Section 4.1.1), and so embeddability on simplicial complexes trivially reduces to embeddability on 2-complexes. Thus, our algorithm actually immediately solves the general problem of embedding graphs on simplicial complexes.

**Reduction from other problems to EMBED .** The EMBED problem unifies several problems presented in the previous chapter:

1. The embeddability of graphs on surfaces, since every surface is a 2-complex.
2. The crossing number problem (see Paragraph 4.2.1). Indeed, let  $k$  be an integer; we define the complex  $\mathcal{C}_k$  as the complex obtained from the sphere by taking  $k$  disjoint open disks and replacing each of them with two isolated segments  $\{u_1, v_1\}$  and  $\{u_2, v_2\}$  such that the endpoints appear along the boundary of the disk in the cyclic order  $u_1, v_1, u_2, v_2$ . See Figure 5.1 on the left for an example for  $k = 3$ . Then, a graph  $G$  is embeddable on  $\mathcal{C}_k$  if and only if its crossing number is at most  $k$ . Moreover, the same construction can be applied on a surface  $\mathcal{S}$  other than the sphere to obtain a complex such that a graph is embeddable on it if and only if it is embeddable on  $\mathcal{S}$  with at most  $k$  crossings. Thus, the EMBED problem generalizes the crossing number problem on arbitrary surfaces.

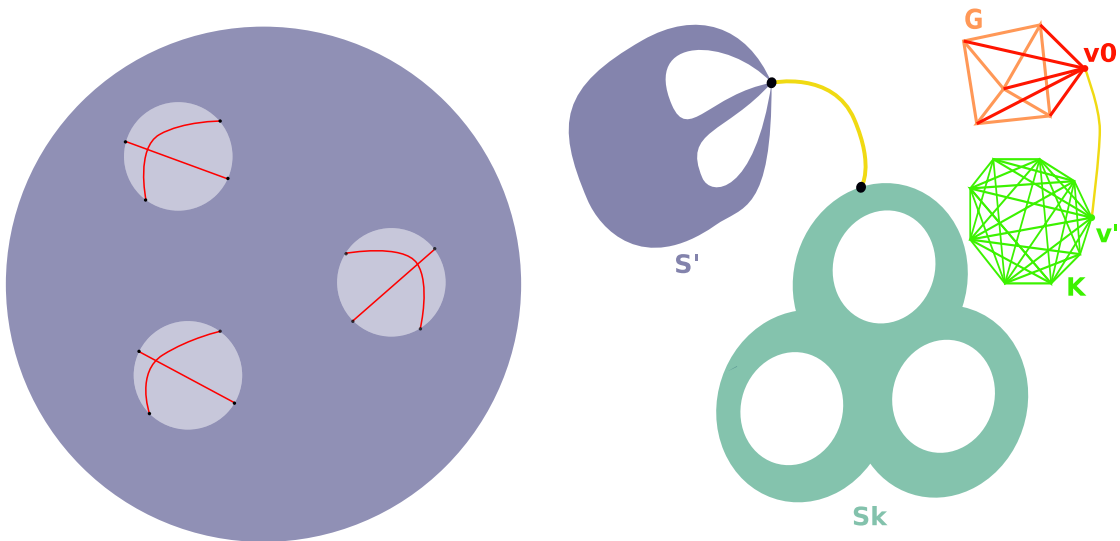


Figure 5.1: On the left: the 2-complex constructed for the crossing number problem for  $k = 3$ . On the right: the 2-complex and the graph constructed for the planarity number problem for  $k = 3$

3. The planarity number of a graph (see Paragraph 4.2.1). Let  $k$  be an integer. We will define a complex  $\mathcal{C}_k$  and, for each input graph  $G$ , a graph  $G'$  such that  $G'$  is

embeddable on  $\mathcal{C}_k$  if and only if  $G$  has planarity number at most  $k$ . Let  $K$  be a 2-connected graph (for all vertices  $v$  of  $K$ ,  $K - v$  is connected) of genus  $2k$  and  $\mathcal{S}_k$  a surface of Euler genus  $2k$  on which  $K$  is embeddable.

First, we define the complex  $\mathcal{C}_k$  (See Figure 5.1 on the right for an example for  $k = 3$ ). Let  $\mathcal{S}$  be a sphere,  $p_1, \dots, p_k$  be  $k$  distinct points of  $\mathcal{S}$  and  $q$  a point of  $\mathcal{S}_k$ . Let  $\mathcal{S}'$  be the complex obtained by identifying all points  $p_1, \dots, p_k$  into a single point  $p$ . Finally, let  $\mathcal{C}_k$  be the complex formed by  $\mathcal{S}'$ ,  $\mathcal{S}_k$  and one isolated segment  $\epsilon$  between  $p$  and  $q$ .

We also define, for each graph  $G$ , the graph  $G'$  obtained from  $G$  and  $K$  by adding a vertex  $v_0$  connected to all vertices of  $G$  and an edge  $e$  between  $v_0$  and an arbitrary vertex  $v'$  of  $K$ . Moreover, later we will define  $G''$  as the subgraph of  $G'$  induced by  $G$  and  $v_0$ .

Then, we have the following lemma:

**Lemma 5.1.1.**  *$G$  have planarity number at most  $k$  if and only if  $G'$  is embeddable on  $\mathcal{C}_k$ .*

This lemma, together with our algorithm of Chapter 7, allows to solve the planarity number problem with input a graph of size  $n$  and a target planarity number  $k$  in time  $2^{O(k^{c_1})}n^{c_2}$  for some constants  $c_1$  and  $c_2$  while Bienstock and Monma [BM88] gave an algorithm solving the planarity number problem with the same input in time  $O(c_0^k n)$  for some other constant  $c_0$ .

Moreover, a similar construction can be applied on a surface  $\mathcal{S}$  of genus  $g$  other than the sphere by replacing  $\mathcal{S}_k$  and  $K$  by respectively a surface of genus  $2k + g$  and a graph of the same genus embeddable on it to obtain an equivalent of the planarity number for other surfaces than the plane.

*Proof of Lemma 5.1.1.* First, if  $G$  has planarity number at most  $k$ , we can construct an embedding  $\Pi_0$  on the sphere  $\mathcal{S}$  where each vertex appears in the facial walk of at least one face of the set of faces  $F$  of size at most  $k$ . Without loss of generality, we can suppose that each face of  $F$  contains at least one point among  $p_1, \dots, p_k$ . When identifying the points  $p_1, \dots, p_k$ , we obtain an embedding  $\Pi_1$  of  $G$  on  $\mathcal{S}'$  where all vertices are incident to the face  $F_0$  containing  $p$ . Then, we embed  $v_0$  on  $p$ , all edges between vertices of  $G$  and  $v_0$  on  $F_0$ ,  $K$  on  $\mathcal{S}_k$  such that  $v'$  is mapped on  $q$  and  $e$  on  $\epsilon$ . This way, we obtain an embedding of  $G'$  on  $\mathcal{C}_k$ .

Conversely, let  $\Pi_k$  be an embedding of  $G_k$  on  $\mathcal{C}_k$  and  $\Pi$  its restriction to  $G$ . First, since  $K$  minus a vertex is connected, the image of  $K$  is either included in  $\mathcal{S}_k$  or in  $\mathcal{S}'$ . Indeed, otherwise there would be two parts of  $K$  only connected by at most one path and thus, by removing one end of this path we would disconnect  $K$ . Moreover, it is easy to see that every graph embeddable on  $\mathcal{S}'$  has genus at most  $2g - 2$  because, intuitively,  $\mathcal{S}'$  can be obtained from the orientable surface of genus  $2g - 2$  by pinching it; see Section 5.2.3 for a similar but more formal reasoning. So, no graph of genus  $2k$  can be embedded on  $\mathcal{S}'$ . So,  $K$  is embedded on  $\mathcal{S}_k$ .



Then, the image of  $G''$  either (1) intersects both  $\mathcal{S}_k$  and  $\mathcal{S}'$ , (2) is included in  $\mathcal{S}_k$ , (3) or is included in  $\mathcal{S}'$ .

Case(1), first  $v_0$  is mapped on  $\mathcal{S}_k$ . Otherwise, there would be at least two edges mapped between  $\mathcal{S}'$  and  $\mathcal{S}_k$ , one connecting  $v_0$  to  $v'$  and one connected  $v_0$  to any vertex of the part of  $G''$  mapped on  $\mathcal{S}_k$ , which is impossible since  $\mathcal{S}'$  and  $\mathcal{S}_k$  are only connected by  $\epsilon$ . For the same reason and since there is an edge between  $v_0$  and each vertex of  $G''$  mapped on  $\mathcal{S}'$ , there is at most one vertex  $u$  on  $\mathcal{S}'$ . Thus, by mapping  $\{u, v_0\}$  and  $u$  on  $\mathcal{S}_k$  on a face incident to  $v_0$ , we can assume that we are in the case (2).

Case (2), since  $\mathcal{S}_k$  is a surface of minimal genus on which  $K$  is embeddable, this embedding is cellular [You63]. Thus,  $G''$  is mapped on a disk and thus  $\Pi$  restricted to  $G$  is a planar embedding where all vertices of  $G$  share a face, the one containing  $v_0$ . So  $G$  obviously has planarity number  $1 \leq k$ .

Case (3),  $e$  goes through  $\epsilon$  since it has one end,  $v_0$ , in  $\mathcal{S}'$  and one,  $v_k$ , in  $\mathcal{S}_k$ . Thus, by shrinking  $\epsilon$  in  $\mathcal{S}'$ , we can suppose that  $v_0$  is mapped on  $p_0$ . Then,  $\Pi$  restricted to  $G$  is an embedding such that all vertices are mapped on the boundary of the face  $F_0$  that contains  $p_0$ . The embedding  $\Pi'$  of  $G$  on  $\mathcal{S}$  corresponding to the one of  $\Pi$  where the points  $p_1, \dots, p_k$  are not tied together. Then,  $\Pi'$  is an embedding of planarity number at most  $k$ . indeed, let  $\mathcal{F}$  be the set of the faces that contains at least one point among  $p_1, \dots, p_k$ .  $\mathcal{F}$  obviously contains at most  $k$  faces. Moreover, all vertices of  $G$  are incident to at least face of  $\mathcal{F}$  since in  $\Pi$  these vertices are incident to the face obtained by identifying points  $p_1, \dots, p_k$  into  $p_0$ .

Thus,  $G'$  is embeddable on  $\mathcal{C}_k$  if and only if  $G$  have planarity number at most  $k$ . □

### 5.1.2 Generality of 2-complexes compared to surfaces

From a topological point of view, 2-complexes are far more general than surfaces. Thus, as we shall see, some topological problems on 2-complexes become significantly easier to solve when restricted to surfaces.

**First hard problem for complexes: homeomorphism.** As seen in Section 3.4.3, two surfaces are homeomorphic if and only if they have the same genus and orientability. Moreover, both the genus and the orientability of a surface can be deduced in linear time from a cellular embedding  $\Pi$  on it. First, the genus can be deduced from  $\Pi$  through the Euler's formula. Moreover, the orientability can be deduced from  $\Pi$  through a traversal of its dual graph.

On the other hand, as seen in Paragraph 4.2.3, deciding whether two complexes are homeomorphic is in **NP** but neither known to be **NP**-complete nor decidable in deterministic polynomial time.

Thus, deciding whether two 2-complexes are homeomorphic is assumed to be harder than deciding whether two surfaces are homeomorphic.

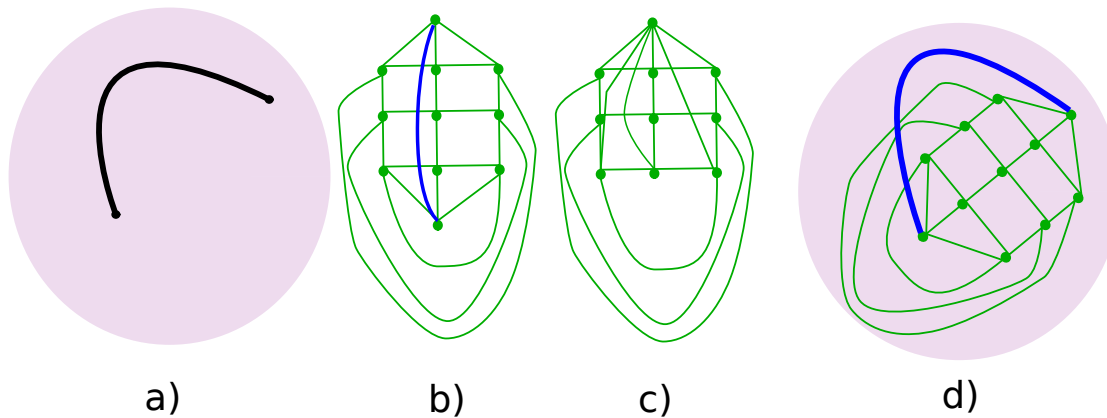


Figure 5.2: 2-complexes and minor closure. a) The target complex: a sphere with one isolated segment. b) A graph  $G_0$ . c) A graph  $H_0$  obtained from  $G_0$  by contracting the blue edge. d) An embedding of  $G_0$  on  $\mathcal{C}_0$ . However,  $H_0$  is not embeddable on  $\mathcal{C}_0$  since, for each edge  $e$  of  $H_0$ ,  $H_0 - e$  is not planar.

**Second hard problem for complexes: contractibility of a closed curve.** The contractibility problem *CONTRACT* is defined as follows: Given a topological space  $X$  and a closed curve  $\ell$  in  $X$ , is  $\ell$  contractible?

The *word problem WORD* is defined as follows: Given a group defined by its finitely many generators and their finitely many relations, and an element  $w$  of the group defined as a word in the generators and their inverses, is  $w$  equal to the unit element of the group? For example, consider the group with generators  $a$  and  $b$  with the relation rules  $a^2b^2 = 1$  and  $b^3 = b^{-1}a^{-1}$ . Is the group element  $a^{-4}b^3ab^{-1}$  equal to 1?

On one hand, F.Lazarus and J.Rivaud [LR12] showed that the contractibility problem restricted to surfaces can be decided in linear time. On the other hand, W.Haken [Hak73] gives a reduction from *WORD* to *CONTRACT* restricted to 2-complexes. Moreover, W.Boone [Boo59] showed that *WORD* is undecidable. Thus, *CONTRACT* restricted to 2-complexes is undecidable.

Thus *CONTRACT* is significantly harder on 2-complexes than on surfaces.

### 5.1.3 The embedding problem: graph minor approach

The set of graphs embeddable on a surface is minor-closed. This leads to an algorithm to decide the embeddability of graphs on surfaces (See Section 4.1.2).

In contrast, the set of graphs embeddable on a complex is not always minor-closed. See Figure 5.2 for an example. This implies that the same approach does not extend to 2-complexes and that Robertson-Seymour theory cannot be applied directly.

Nevertheless, the following weaker property holds. For any given 2-complex  $\mathcal{C}$  either:

- All graphs are embeddable on  $\mathcal{C}$  when  $\mathcal{C}$  contains a 3-book (see Section 5.2.1)
- There is a surface  $\mathcal{S}$  such that any graph embeddable on  $\mathcal{C}$  is also embeddable

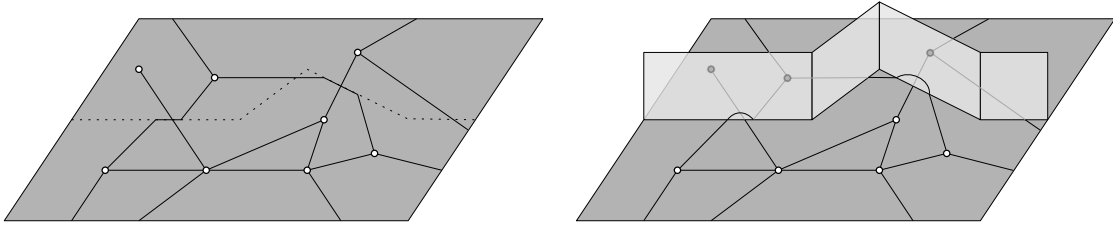


Figure 5.3: Illustration of the proof of Proposition 5.2.1. Left: The drawing of the graph  $G$  and the dotted curve  $c$ . Right: The construction of the 3-book and the modification of the drawing.

on  $\mathcal{S}$  (see Section 5.2.3). Then, if  $K$  is a forbidden minor of  $\mathcal{S}$  then any graph admitting  $K$  as a minor is not embeddable on  $\mathcal{C}$ .

## 5.2 Basic tools

In this section, we will present some basic structures on 2-complexes that will be useful in the study of EMBED .

### 5.2.1 3-books

A 3-book is the 2-complex formed of three triangles incident to one same edge. First, we show the following property:

**Proposition 5.2.1.** *All graphs are embeddable on a complex containing a 3-book.*

*Proof.* Let  $G$  be a graph. We first draw  $G$ , possibly with crossings, in general position in the interior of a closed disk  $D$ . Let  $c$  be a simple curve in  $D$  with endpoints on  $\partial D$  the frontier of  $D$  and passing through all crossing points of the drawing of  $G$ . By perturbing  $c$ , we can ensure that, in the neighborhood of each crossing point of that drawing,  $c$  coincides with the image of one of the two edges involved in the crossing. See Figure 5.3, left.

Let  $D'$  be a closed disk disjoint from  $D$ . We attach  $D'$  to  $D$  by identifying  $c$  with a part of the boundary of  $D'$  and obtain a complex homeomorphic to a 3-book. Now, in the neighborhood of each crossing of the drawing of  $G$ , we push inside  $D'$  the part of the edge coinciding with  $c$ , keeping its endpoints fixed. See Figure 5.3, right. This removes the crossings.

So  $G$  embeds in the topological representation of a 3-book. Thus,  $G$  embeds on  $\mathcal{C}$  since  $\mathcal{C}$  contains a 3-book.  $\square$

All graphs being embeddable on a complex containing a 3-book, **in the rest of the thesis we will only consider complexes containing no 3-book.**

### 5.2.2 Structures on vertices of 2-complexes without 3-books

First, a vertex incident to no edge is an *isolated point* and an edge incident to no triangles is an *isolated segment*.

**Link of a vertex** Let  $\mathcal{C}$  be a 2-complex without 3-book and  $p$  a vertex of  $\mathcal{C}$ .

Intuitively, the link of  $p$  is the graph forming the frontier separating it from the rest of the complex. Formally:

**Definition 5.2.2.** *The link of  $p$  (in  $\mathcal{C}$ ) is the graph  $(V, E)$  where:*

- $V$  is the set of vertices  $u$  such that  $\{p, u\}$  is an edge of  $\mathcal{C}$ .
- $E$  is the set of edges  $\{u, v\}$  such that  $\{p, u, v\}$  is a triangle of  $\mathcal{C}$ .

The link of a vertex will only be used to define the following notion of link component of a vertex.

**Link components of a vertex** Let  $\mathcal{C}$  be a 2-complex without 3-book and  $p$  a vertex of  $\mathcal{C}$ .

Intuitively, a link component of  $p$  is the part of  $\mathcal{C}$  comprised between  $p$  and a connected component of the link of  $p$ . Formally:

**Definition 5.2.3.** *Let  $\mathcal{C}'$  be a subcomplex of  $\mathcal{C}$ . Then,  $\mathcal{C}'$  is a link component of  $p$  if it exists a connected component  $(V_0, E_0)$  of the link of  $p$  such that:*

- The vertices of  $\mathcal{C}'$  are  $V_0 \cup \{p\}$ .
- The edges of  $\mathcal{C}'$  are  $E_0 \cup \{\{p, q\} \mid q \in V_0\}$ .
- The triangles of  $\mathcal{C}'$  are  $\{\{p, q, r\} \mid \{q, r\} \in E_0\}$ .

We can see that any vertex  $u$  of the link  $p$  of degree  $k$  correspond to an edge  $\{p, u\}$  incident to  $k$  triangles of  $\mathcal{C}$ . So, since the  $\mathcal{C}$  contains no 3-book, all vertices of the link of  $p$  have degree at most two. Thus, a connected component of the link of  $p$  can have three forms: a single vertex, a cycle, or a simple path with two distinct endpoints. Let  $C_p$  be a link component of  $p$ . There are three possibilities for the nature of  $C_p$  depending on the type of connected component of the link associated with it. In each case we will present the shape of  $C_p$ :

- The link corresponding to  $C_p$  is a single vertex. Then, the corresponding link component is an isolated segment and its topological representation is as a simple curve.
- The link corresponding to  $C_p$  is a cycle. Then  $C_p$  is a *cone* and its topological representation is homeomorphic to a closed disk with  $p$  inside.
- The link corresponding to  $C_p$  is a simple path with two distinct endpoints. Then  $C_p$  is a *corner* and its topological representation is homeomorphic to a closed disk with  $p$  on the boundary.

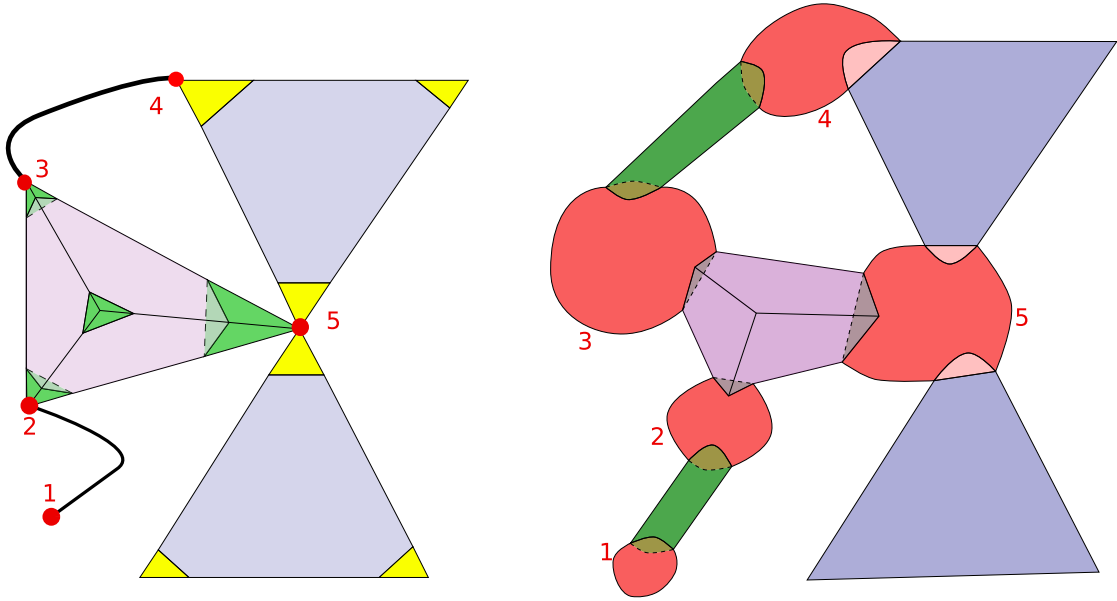


Figure 5.4: On the left: the complex  $\mathcal{C}$  of Figure 1.4 with 5 singular points, numbered from 1 to 5, and 2 isolated segments (one between 3 and 4 and one between 1 and 2) where the cones are in green and the corners in yellow. On the right: A supersurface of  $\mathcal{C}$  where spheres and cylinders replacing respectively singular points and isolated segments of  $\mathcal{C}$  are respectively in red and green.

**Singular points** Intuitively, a singular point is a vertex where the complex is not locally a surface.

Formally, we define a singular point as follows:

**Definition 5.2.4.** *Let  $\mathcal{C}$  be a complex and  $p$  a vertex of  $\mathcal{C}$ . Then,  $p$  is a singular point if either  $p$  is incident to no triangle or there is several link components of  $p$  (the link of  $p$  is not connected). Otherwise, when  $p$  is not a singular point, we say that  $p$  is a regular point*

Surfaces, possibly disconnected, possibly with boundaries, are exactly 2-complexes without singular points.

**Lemma 5.2.5.** *Let  $\mathcal{C}$  be a 2-complex containing no 3-book. Then,  $\mathcal{C}$  is a surface, possibly disconnected, possibly with boundaries, if and only if it contains no singular points.*

*Proof.* Let  $\mathcal{C}$  be a 2-complex containing neither a singular point nor a 3-book. First,  $\mathcal{C}$  contains no isolated segment. Otherwise, let  $p$  and  $q$  be the endpoints of an isolated segment. Then,  $q$  is a vertex of degree 0 of the link of  $p$ . So, if the link of  $p$  contains at least 2 vertices, its link is not connected. If the link of  $p$  contains only  $q$ ,  $p$  is not incident to any other edge and thus not incident to any triangle. Thus, in both case,  $p$  is a singular point.

Then, we will show that any point of  $\mathcal{C}$  have a neighborhood homeomorphic to a disk.

First, let  $p$  be a vertex of  $\mathcal{C}$ . The link of  $p$  is connected. Thus,  $p$  have only one link component  $C_p$ . Moreover,  $C_p$  is either a corner component in which case its regular neighborhood is homeomorphic to a half-disk or a disk component in which case its regular neighborhood is homeomorphic to a disk (it cannot be an isolated segment). So, if  $\mathcal{C}$  contains no singular points, all vertices of  $\mathcal{C}$  have a neighborhood homeomorphic to either a disk or a half-disk

Moreover, any point of an edge incident to one or two triangles admits a neighborhood homeomorphic to respectively a half-disk or a disk. Thus, since  $\mathcal{C}$  contains neither a 3-book nor an isolated segment, all points of an edge of  $\mathcal{C}$  have a neighborhood homeomorphic to a disk.

Furthermore, a point inside a triangle admits the triangle as a neighborhood homeomorphic to a disk.

Thus, if  $\mathcal{C}$  contains neither a singular point nor a 3-book, then all points of  $\mathcal{C}$  admit a neighborhood homeomorphic to either a disk or a half-disk and thus is a surface .

Conversely, let  $p$  be a vertex of a surface possibly disconnected, possibly with boundary  $S$ . Then,  $p$  admits a neighborhood  $N$  homeomorphic to either a disk or a half-disk. First, in both case,  $N$  intersects at least one triangle which is obviously incident to  $p$ .  $N$  intersecting all triangles and edges incident go  $p$ , we can see that the link of  $p$  is either a simple path when  $N$  is a half disk or a cycle when  $N$  is homeomorphic to a disk. In both cases, the link of  $p$  is connected. Thus,  $p$  is not a singular point.

□

### 5.2.3 Supersurface

Let  $\mathcal{C}$  be a 2-complex containing no 3-book. We define a surface possibly disconnected, possibly with boundary called a *supersurface of  $\mathcal{C}$*  such that all graphs embeddable on  $\mathcal{C}$  are also embeddable on its supersurface. Intuitively, the supersurface is the surface obtained from the complex by thickening all isolated segments and all singular points.

See Figure [5.4](#) for an example of supersurface.

Formally, a supersurface is defined as follows:

First, we replace every isolated vertex of  $\mathcal{C}$  with a sphere and every isolated segment of  $\mathcal{C}$  with a cylinder. Then, for every singular point  $p$ , we do the following. We remove a small neighborhood of  $p$ . We create a sphere with  $k$  boundary components, where  $k$  is the number of link components at  $p$ . Finally, we attach a link component to each of the  $k$  boundary components of the sphere:

- for each link component that was a cone, a small neighborhood of  $p$  was removed, with boundary a circle; we attach that circle bijectively to a boundary component of the sphere;
- for each link component that was a corner, a small neighborhood of  $p$  was removed, with boundary a segment; we attach that segment to a part of a boundary component of the sphere;

- for each link component that was an isolated segment, the isolated segment was replaced with a cylinder; we attach the corresponding boundary component of that cylinder bijectively to a boundary component of the sphere.

(Remark: There exist several surfaces that can be obtained this way depending on the orientations chosen for the paths and cycles we glue together.)

A supersurface of a 2-complex have the following property, the proof of which should be intuitively clear.

**Proposition 5.2.6.** *All graphs embeddable on a 2-dimensional complex  $\mathcal{C}$  are also embeddable on  $\mathcal{S}$ , a supersurface of  $\mathcal{C}$ .*

*Proof.* Consider an embedding of a graph  $G$  on  $\mathcal{C}$ . It is not hard to transform that embedding into an embedding of  $G$  on  $\mathcal{S}$ : Each cylinder replacing an isolated segment is used only along a single path connecting its two boundary components; if a singular point  $p$  is used by the embedding of  $G$  on  $\mathcal{C}$ , we can locally modify the embedding to accommodate the local change from  $\mathcal{C}$  to  $\mathcal{S}$  at  $p$  (see again Figure 5.4).

Thus, all graphs embeddable on  $\mathcal{C}$  are also embeddable on  $\mathcal{S}$ . □

In particular, the graphs embeddable on a fixed 2-complex have bounded genus.

# Chapter 6

## Embedding graphs on 2-complexes: a first algorithm

The contents of this chapter, written jointly with Éric Colin de Verdière and Bojan Mohar, appeared in Proceedings of the 34th International Symposium on Computational Geometry (SoCG 2018) [CMM18].

### 6.1 Preliminaries

#### 6.1.1 Result of this chapter

In this chapter we study the *EMBED* problem: given a graph  $G$  and a 2-complex  $\mathcal{C}$ , decide whether there exists an embedding of  $G$  on  $\mathcal{C}$ . We present a **XP** algorithm with parameter the input complex (meaning that the algorithm works in polynomial time when the input complex is fixed) deciding *EMBED*: given a graph  $G$  and a 2-complex  $\mathcal{C}$ , decide whether there exists an embedding of  $G$  on  $\mathcal{C}$ . Simultaneously, we will also prove that the problem *EMBED* is in NP. Formally, the two results of this chapter are the following:

**Theorem 6.1.1.** *The problem *EMBED* is NP-complete.*

**Theorem 6.1.2.** *The problem *EMBED* can be solved in time  $f(c) \cdot n^{O(c)}$ , where  $n$  is the number of edges and vertices of the input graph,  $c$  is the number of simplices and  $f$  is some computable function of  $c$ .*

As for Theorem 6.1.1, it is straightforward that the problem is NP-hard (as the case where  $\mathcal{C}$  is a surface is already NP-hard); the interesting part is to provide a certificate checkable in polynomial time when an embedding exists. Note that Theorem 6.1.2 shows that, for every fixed complex  $\mathcal{C}$ , the problem of deciding whether an input graph embeds into  $\mathcal{C}$  is polynomial-time solvable. Actually, our algorithm is explicit, in the sense that, if there exists an embedding of  $G$  on  $\mathcal{C}$ , we can provide some representation of such



an embedding (in contrast to various results in the theory of graph minors, where the existence of an embedding can be obtained without leading to an explicit construction). (In the rest of the chapter, we use the parameters  $n$  and  $c$  whenever we need to refer to the input size. Moreover, to avoid to confound the 1-skeleton of  $\mathcal{C}$  with  $G$ , when necessary, we will refer to the vertices and edges of  $\mathcal{C}$  as respectively nodes and arcs)

### 6.1.2 Strategy of the proof and organization of the chapter

First, recall that every graph is embeddable on a 3-book (Section 5.2.1). So, without loss of generality, in this chapter we will assume that the input 2-complex  $\mathcal{C}$  contains no 3-book.

The idea of the algorithm is to progressively reduce the problem to simpler problems. Thus, we reduce EMBED to *embedding extension problems* (EEP), similar to the EMBED problem except that an embedding of a subgraph  $H$  of the input graph  $G$  is already specified. In Section 6.2, we reduce EMBED to EEPs on a pure 2-complex (a 2-complex without isolated segments). In Section 6.3, we further reduce it to EEPs on a surface. In Section 6.4, we reduce it to EEPs on a surface in which every face of the subgraph  $H$  is a disk. Finally, in Section 6.5, we show how to solve EEPs of the latter type using a key component in an algorithm by B.Mohar [Moh99] to decide embeddability of a graph on a surface.

### 6.1.3 Embedding extension problems and reductions

An *embedding extension problem* (EEP) is a decision problem which asks if an embedding of a subgraph can be extended to an embedding of the whole graph in the same complex  $\mathcal{C}$ . Formally, we will parameterize it with the size of the graph, the size of the complex  $\mathcal{C}$  and with a third parameter which will describe the "topological size" of the pre-embedded subgraph. For the latter one we need a concept of a branch.

Let  $H$  be a graph. A simple path  $P$  with vertices  $u_1 u_2 \dots u_k$  or a cycle  $C$  with vertices  $u_1 u_2 \dots u_{k-1} u_1$  ( $k \geq 2$ ) in  $H$  is a *branch* of  $H$  if  $\deg_H(u_1) \neq 2$ ,  $\deg_H(u_k) \neq 2$ , and  $\deg_H(u_i) = 2$  for every  $i = 2, \dots, k-1$ . If  $H$  has a component isomorphic to a cycle, we consider such a component as a branch as well. Under this convention we treat an isolated vertex of  $H$  as a branch as well. Then every graph is the edge-disjoint union of its branches.

**EEP**( $n, m, c$ ):

INPUT: A graph  $G$  with at most  $n$  vertices and edges, a subgraph  $H$  of  $G$  with at most  $m$  branches, and an embedding  $\Pi$  of  $H$  into a 2-complex  $\mathcal{C}$  with at most  $c$  simplices.

QUESTION: Does  $G$  have an embedding into  $\mathcal{C}$  whose restriction to  $H$  is  $\Pi$ ?

To be precise, we will have to explain how we represent the embedding  $\Pi$ , but this will vary throughout the proof, and we will be more precise about this in subsequent sections. Let us simply remark that, since the complexity of our algorithm is a polynomial of large degree (depending on the complex  $\mathcal{C}$ ) in the size of the input graph, the choice of representation is not very important, because converting between any two reasonable representations is possible in polynomial time.

We will reduce our original problem to more and more specialized EEPs. We will use the word "reduce" in a somewhat sloppy sense: A decision problem  $P$  *reduces* to  $k$  instances of the decision problem  $P'$  if solving these  $k$  instances of  $P'$  allows to solve the instance of  $P$  in time  $O(k)$ .

## 6.2 Reduction to EEPs on a pure 2-complex

Our first task is to reduce the problem EMBED to a problem on a pure 2-dimensional complex. More precisely, let **EEP-SING** be the problem EEP, restricted to instances  $(G, H, \Pi, \mathcal{C})$  where:  $\mathcal{C}$  is a pure 2-complex containing no 3-books;  $H$  is a set of vertices of  $G$ ; and  $\Pi$  is an injective map from  $V(H)$  to the nodes of  $\mathcal{C}$  such that  $\Pi(H)$  contains all singular points of  $\mathcal{C}$ . In this section, we prove the following result.

**Proposition 6.2.1.** *Any instance of EMBED  $(n, c)$  reduces to  $(cn)^{O(c)}$  instances of EEP-SING  $(cn, c, O(c))$ .*

First, a definition. Consider a map  $f : P \rightarrow V(G) \cup \{\varepsilon\}$ , where  $P$  is a set of nodes in  $\mathcal{C}$  containing all singular points of  $\mathcal{C}$ . We say that an embedding  $\Gamma$  of  $G$  in  $\mathcal{C}$  *respects*  $f$  if, for each  $p \in P$ , the following holds: If  $f(p) = \varepsilon$ , then  $p$  is not in the image of  $\Gamma$ ; otherwise,  $\Gamma(f(p)) = p$ .

In this section, we will need the following intermediate problem:

**EMBED-RESP** $(n, m, c)$ :

INPUT: A graph  $G$  with at most  $n$  vertices and edges, a 2-complex  $\mathcal{C}$  (not necessarily pure) with at most  $c$  simplices that contains no 3-books, and a map  $f$  as above with domain of size at most  $m$ .

QUESTION: Does  $G$  have an embedding into  $\mathcal{C}$  respecting  $f$ ?

**Lemma 6.2.2.** *Any instance of EMBED  $(n, c)$  reduces to  $(O(cn))^c$  instances of EMBED-RESP  $(cn, c, c)$ .*

*Proof.* By Proposition [5.2.1](#), we can without loss of generality assume that  $\mathcal{C}$  contains no 3-books. Let  $G'$  be the graph obtained from  $G$  by subdividing each edge  $k$  times, where  $k \leq c$  is the number of singular points of  $\mathcal{C}$ . We claim that  $G$  has an embedding into  $\mathcal{C}$  if and only if  $G'$  has an embedding  $\Gamma'$  into  $\mathcal{C}$  such that each singular point of  $\mathcal{C}$  in the image of  $\Gamma'$  is the image of a vertex of  $G'$ .

Indeed, assume that  $G$  has an embedding  $\Gamma$  on  $\mathcal{C}$ . Each time an edge of  $G$  is mapped, under  $\Gamma$ , to pass through a singular point  $p$  of  $\mathcal{C}$ , we subdivide this edge and map this new vertex to  $p$ ; the image of the embedding is unchanged. This ensures that only vertices are mapped to singular points. Moreover, there were at most  $k$  subdivisions, one per singular point. So, by further subdividing the edges until each original edge is subdivided  $k$  times, we obtain an embedding of  $G'$  to  $\mathcal{C}$  such that only vertices are mapped on singular points. The reverse implication is obvious: If  $G'$  has an embedding into  $\mathcal{C}$ , then so has  $G$ . This proves the claim.

To conclude, for each map from the set of singular vertices of  $\mathcal{C}$  to  $V(G') \cup \{\varepsilon\}$ , we solve the problem whether  $G'$  has an embedding on  $\mathcal{C}$  respecting  $f$ . The graph  $G$

embeds on  $\mathcal{C}$  if and only if the outcome is positive for at least one such map  $f$ . By construction, there are at most  $(kn + 1)^k = (O(cn))^c$  such maps, because  $V(G')$  has size at most  $kn$ .  $\square$

**Lemma 6.2.3.**  $\text{EMBED-RESP}(n, m, c)$  reduces to  $\text{EEP-SING}(n, m, O(c))$ .

*Proof.* Formally, we describe a set of transformations on  $\mathcal{C}$ ,  $G$ , and  $f$ . The invariant is that they preserve the existence or non-existence of an embedding of  $G$  into  $\mathcal{C}$  respecting  $f$ .

*Step 1.* We start by dissolving all degree-two vertices of  $G$  that are not in the image of  $f$ . (By *dissolving a degree-two vertex*  $v$ , incident to edges  $vv_1$  and  $vv_2$ , we mean removing  $v$  and replacing  $vv_1$  and  $vv_2$  with a single edge  $v_1v_2$ .) It is clear that the original graph  $G$  has an embedding in  $\mathcal{C}$  respecting  $f$  if and only if the new graph (still called  $G$ ) has an embedding in  $\mathcal{C}$  respecting  $f$ .

*Step 2.* If a singular point is not used, then we can remove it without affecting the embeddability of  $G$ . However, removing a vertex from a 2-complex does not yield a 2-complex. We thus define a 2-complex that has the same properties. Let  $p$  be a singular point of a 2-complex  $\mathcal{C}$ . Let  $T$  be the set of triangles and segments of  $\mathcal{C}$  incident with  $p$ , uniquely partitioned into  $T_1, \dots, T_k$ , where each  $T_i$  is either a cone, a corner, or an isolated segment. The *withdrawal of  $p$  from  $\mathcal{C}$*  is the complex obtained by doing the following operation for each  $i = 1, \dots, k$ : We first create a new node  $p_i$ , and then replace, in each triangle and edge of  $T_i$ , the node  $p$  by  $p_i$ .

For every singular point  $p$  of  $\mathcal{C}$  such that  $f(p) = \varepsilon$  and incident to at least two segments, we withdraw  $p$  from  $\mathcal{C}$ . For each created node  $p_i$ , we let  $f(p_i) = \varepsilon$ . The fact that  $G$  embeds, or not, on  $\mathcal{C}$  respecting  $f$  is preserved: Indeed, if  $G$  embeds on the original complex respecting  $f$ , then this corresponds to an embedding on the complex obtained by withdrawing  $p$ , and avoiding  $p_1, \dots, p_k$ , thus respecting  $f$ ; conversely, if  $G$  embeds on the complex obtained by withdrawing  $p$ , respecting  $f$ , then it avoids the  $p_i$ s, and, after identifying together the points  $p_i$ ,  $1 \leq i \leq k$ , to a single point  $p$ , this embedding avoids  $p$ , and thus respects  $f$ .

*Step 3.* At this point, every singular point  $p$  with  $f(p) = \varepsilon$  is incident to exactly one segment, and to no triangle. For each isolated segment  $pq$  of  $\mathcal{C}$  such that  $u := f(p)$  and  $v := f(q)$  are both different from  $\varepsilon$ , and  $G$  contains an edge  $uv$  connecting  $u$  and  $v$ , we remove  $uv$  from  $G$  and remove  $pq$  from  $\mathcal{C}$ . We need to prove that this operation does not affect the (non-)existence of an embedding of  $G$  respecting  $f$ . First, assume that, initially, there was an embedding  $\Gamma$  of  $G$  on  $\mathcal{C}$  respecting  $f$ ; then either segment  $pq$  is used by  $uv$ , in which case clearly there is still an embedding after this operation, or edge  $uv$  does not use segment  $pq$  at all, in which case we can first modify  $\Gamma$  by embedding edge  $uv$  on segment  $pq$  and by moving the edges on  $pq$  on the space where  $uv$  was before, so we are now in the previous case. Conversely, if after this operation  $G$  has an embedding on  $\mathcal{C}$  respecting  $f$ , trivially it is also the case before.

*Step 4.* For each isolated segment  $pq$  of  $\mathcal{C}$  such that  $u := f(p)$  and  $v := f(q)$  are both different from  $\varepsilon$ , but  $G$  contains no edge connecting  $u$  and  $v$ , we do the following. In  $\mathcal{C}$ , we remove  $pq$  and add a new segment  $p'q'$  where  $p'$  and  $q'$  are new nodes; we also

extend  $f$  by letting  $f(p') = f(q') = \varepsilon$ . Finally, if  $G$  contains at least one edge of the form  $ux$ , where  $x$  has degree one and is not in the image of  $f$ , we remove a single such edge; similarly, if it contains at least one edge of the form  $vy$ , where  $y$  has degree one and is not in the image of  $f$ , we remove a single such edge. This operation does not affect our invariant, for similar reasons; for example, if initially there was an embedding of  $G$  respecting  $f$ , then segment  $uv$  can only contain edges of  $G$  that are themselves connected components of  $G$ , which we can re-embed on the new segment  $p'q'$ , or edges of the form  $ux$  or  $vy$ , where  $x$  and  $y$  have degree one and are not in the image of  $f$ .

*Step 5.* For each isolated segment  $pq$  where  $u := f(p)$  is different from  $\varepsilon$  but  $f(q) = \varepsilon$ , we remove  $pq$  from  $\mathcal{C}$  and add a new segment  $p'q'$  where  $p'$  and  $q'$  are new nodes; we also extend  $f$  by letting  $f(p') = f(q') = \varepsilon$ . Finally, if  $G$  contains at least one edge of the form  $ux$ , where  $x$  has degree one and not in the image of  $f$ , we remove a single such edge. The invariant is preserved, for reasons similar to the previous case.

*Step 6.* Now, every segment of the complex (still called  $\mathcal{C}$ ) is incident to one or two triangles, except perhaps some segments that are themselves connected components of  $\mathcal{C}$  and whose endpoints are not in the domain of  $f$ . If there is at least one such segment, we remove all of them from  $\mathcal{C}$ , and remove all edges  $uv$  from  $G$  that are themselves connected components of  $G$  and such that  $u$  and  $v$  are not in the image of  $f$ ; as above, this does not affect whether  $G$  embeds into  $\mathcal{C}$  respecting  $f$ .

*Step 7.* Finally, for each node  $p$  of  $\mathcal{C}$  incident to no segment, such that  $u := f(p)$  is different from  $\varepsilon$ , we do the following: If  $u$  has degree zero, we remove  $p$  and  $u$ ; otherwise, we immediately return that  $G$  does not embed into  $\mathcal{C}$  respecting  $f$ . For each node  $p$  of  $\mathcal{C}$  that is incident to no segment and such that  $f(p) = \varepsilon$ , we remove  $p$ , and remove a single degree-zero vertex of  $G$  not in the image of  $f$ , if one exists.

*Conclusion.* Now,  $\mathcal{C}$  has no node that is itself a connected component; each of its segments is incident to one or two triangles. Also,  $f$  maps each singular point of  $\mathcal{C}$  to a vertex of  $G$ . It may map some other nodes of  $\mathcal{C}$  to  $\varepsilon$ , but such nodes are non-singular and, if an embedding uses them, a slight perturbation will avoid them, so we can remove the nodes  $p$  such that  $f(p) = \varepsilon$  from the domain of  $f$  without affecting the result. Now, we have an EEP as specified in the statement of Proposition [6.2.1](#).

Finally, it is easy to check that, in each of the seven steps above, the numbers of vertices and edges of  $G$  do not increase, and the number of simplices of  $\mathcal{C}$  increase by at most a multiplicative constant. Moreover, the domain of  $f$  also does not increase (it increases when we withdraw singular points, but the images of the new nodes are  $\varepsilon$ , and such nodes are later removed from the domain of  $f$ ).  $\square$

*Proof of Proposition [6.2.1](#).* It follows immediately from Lemmas [6.2.2](#) and [6.2.3](#).  $\square$

### 6.3 Reduction to an EEP on a possibly disconnected surface

The previous section led us to an embedding extension problem in a pure 2-complex without 3-books where the images of some vertices are predetermined. Now, we show

that solving such an EEP amounts to solving another EEP in which the complex is a surface.

Let **EEP-SURF** be the problem EEP, restricted to instances where the input complex is (homeomorphic to) a possibly disconnected triangulated surface without boundary (which we denote by  $\mathcal{S}$  instead of  $\mathcal{C}$ , for clarity). To represent the embedding  $\Pi$  in such an EEP instance  $(G, H, \Pi, \mathcal{S})$ , it will be convenient to use the fact that, in all our constructions below, the image of every connected component of  $H$  under  $\Pi$  will intersect the 1-skeleton of  $\mathcal{S}$  at least once, but only in a finite number of points. (Note that  $H$  may use some nodes of  $\mathcal{S}$ .) Consider the *overlay* of the triangulation of  $\mathcal{S}$  and of  $\Pi$ , the union of the 1-skeleton of  $\mathcal{S}$  and of the image of  $\Pi$ ; this overlay is the image of a graph on  $\mathcal{S}$ ; each of its edges is either a piece of the image of an edge of  $H$  or a piece of a segment of  $\mathcal{S}$ ; each of its vertices is the image of a vertex of  $H$  and/or a node of  $\mathcal{S}$ . By the assumption above on  $\Pi$ , this overlay is cellularly embedded on  $\mathcal{S}$ , and we can represent it by its combinatorial map [Lin82; Epp03] (possibly on surfaces with boundary, since at intermediary steps of our construction we will have to consider such surfaces).

In this section, we prove the following proposition.

**Proposition 6.3.1.** *Any instance of EEP-SING( $n, m, c$ ) reduces to an instance of EEP-SURF( $O(n + m + c), O(m + c), O(c)$ ).*

We will first reduce the original EEP to an intermediary EEP on a surface with boundary.

**Lemma 6.3.2.** *Any instance of EEP-SING( $n, m, c$ ) reduces to an instance of EEP( $n + O(c), m + O(c), O(c)$ ) in which the considered 2-complex is a possibly disconnected surface with boundary.*

*Proof.* The key property that we will use is that, since  $\mathcal{C}$  is pure and contains no 3-books, each singular point is incident to cones and corners only.

Figure 6.1 illustrates the proof. Let  $(G, H, \Pi, \mathcal{C})$  be the instance of EEP-SING. We first describe the construction of the instance  $(G', H', \Pi', \mathcal{S})$  where  $\mathcal{S}$  is a supersurface of  $\mathcal{C}$  (see Section 5.2.3).

We define  $H', G'$ , and  $\Pi'$  from  $H, G$ , and  $\Pi$  (again, refer to Figure 6.1). Let  $p$  be a singular point of  $\mathcal{C}$  and  $v_p$  the vertex of  $H$  mapped on  $p$  by  $\Pi$ . Let  $c_p$  be the number of cones at  $p$  and  $c'_p$  be the number of corners at  $p$ . In  $H$  (and thus also  $G$ ), we add a set  $L_p$  of  $c_p + c'_p$  loops with vertex  $v_p$ . Let  $q_p$  be a point in the interior of  $S_p$ ; in  $\Pi$ , we map  $v_p$  to  $q_p$ , and we map these  $c_p + c'_p$  loops on  $S_p$  in such a way that each loop encloses a different boundary component of  $S_p$  (thus, if we cut  $S_p$  along these loops, we obtain  $c_p + c'_p$  annuli and one disk).

Finally, we add to  $H$  (and thus also to  $G$ ) a set  $E_p$  of  $c'_p$  new edges, each connecting  $v_p$  to a new vertex. In  $\Pi$ , each new vertex is mapped to the boundary component of  $S_p$  corresponding to a corner, but not on the corresponding arc.

Let us call  $G'$  and  $H'$  the resulting graphs, and  $\Pi'$  the resulting embedding of  $H'$ . Note that, from the triangulation of  $\mathcal{C}$  with  $c$  simplices, we can easily obtain a triangulation

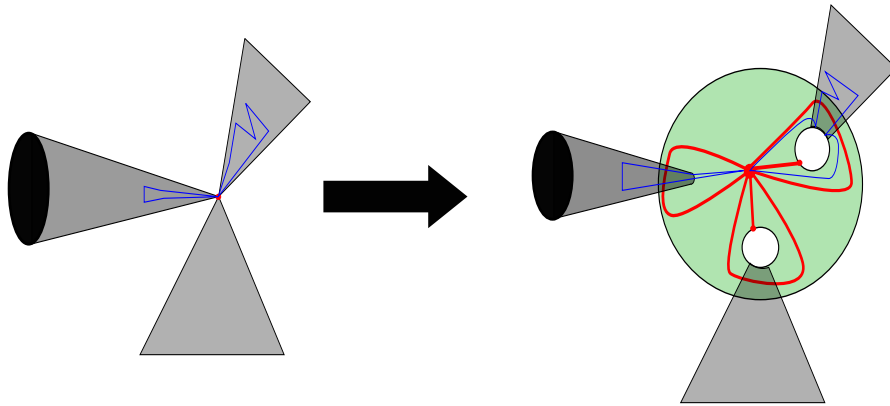


Figure 6.1: The modification of singular vertices in the proof of Lemma [6.3.2](#). We transform the neighborhood of each singular vertex to make it surface-like. Moreover, we add to  $H$  one loop per cone or corner, and for each corner, we add a vertex and an edge.

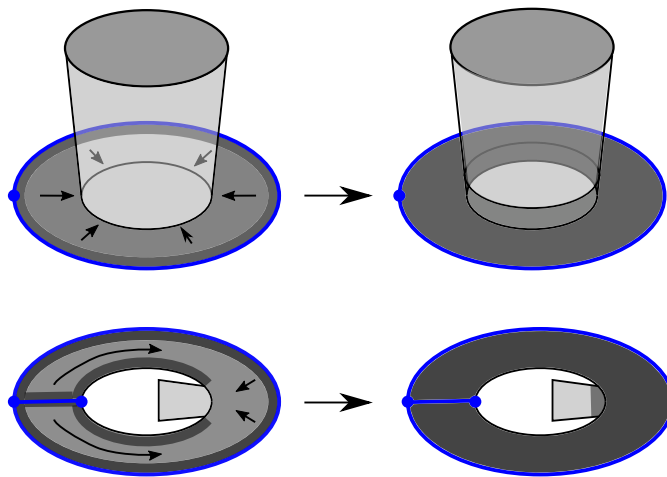


Figure 6.2: In the proof of Lemma [6.3.2](#), we push some parts of the graph outside  $S_p$  by an ambient isotopy of  $\mathcal{S}$  that moves a small enough open disk or annulus (in blue), which is disjoint from  $L_p$  and  $E_p$ , to a larger part.

of  $\mathcal{S}$  with  $O(c)$  simplices, and that the image of each edge of  $H$  crosses  $O(1)$  edges of this triangulation.

It remains to prove that the two EEPs constructed are equivalent.

Let us first prove that any solution  $\Gamma$  of the EEP instance  $(G, H, \Pi, \mathcal{C})$  yields a solution of the EEP instance  $(G', H', \Pi', \mathcal{S})$ . Let  $p$  be a singular point of  $\mathcal{C}$ , and let  $v_p$  be the vertex of  $H$  mapped to  $p$ . We need to modify  $\Gamma$  locally in the neighborhood  $N_p$  of  $p$  that is removed when transforming  $\mathcal{C}$  to  $\mathcal{S}$ . Without loss of generality, up to an ambient isotopy of  $\Gamma$  that does not move  $H$ , we can assume that the image of  $\Gamma$  intersects  $N_p$  exactly in straight line segments having  $p$  as one endpoint. To build a solution of  $(G', H', \Pi', \mathcal{S})$ , we first remove the part of  $\Gamma$  inside  $N_p$ , and we reconnect  $v_p$  to each point of the image of  $\Gamma$  lying on the boundary of  $N_p$ , by simple paths on  $S_p$ ; this is certainly possible because each point of  $S_p$  not in the image of  $E_p \cup L_p$  can be connected to  $v_p$  by a simple path that does not meet  $E_p \cup L_p$  (except at  $v_p$ ). Thus,  $(G', H', \Pi', \mathcal{S})$  has a solution.

Conversely, let  $\Gamma'$  be a solution of the EEP  $(G', H', \Pi', \mathcal{S})$ ; we build a solution of  $(G, H, \Pi, \mathcal{C})$ . As above, let  $q_p$  be a point of  $\mathcal{S}$  that was obtained from a singular point  $p$  of  $\mathcal{C}$ . We now show that we can assume that  $\Gamma$  does not enter  $S_p$ , except for  $v_p, E_p, L_p$ , and the edges in  $G'$  incident to  $v_p$ .

First, the part of  $\Gamma$  lying in the connected component  $D_p$  of  $S_p$  minus the image of  $L_p$  that is a disk corresponds to a planar subgraph of  $G$ , connected to the rest of  $G$  by  $v_p$  only; we can re-embed this planar subgraph of  $G$  in  $S_p \setminus D_p$ . Next, consider a cone at  $p$ . The part of  $S_p$  enclosed by the corresponding loop of  $L_p$  is an annulus; the situation is as on Figure 6.2, top, and, by an ambient isotopy of  $\mathcal{S}$ , we can push the part of  $\Gamma$  that lies in the annulus outside it, except for those edges touching  $v_p$ . Finally, consider a corner at  $p$ , and the annulus that is the part of  $S_p$  enclosed by the corresponding loop in  $L_p$ . The local picture for this part of  $S_p$  is as shown on Figure 6.2, bottom, and similarly by an ambient isotopy of  $\mathcal{S}$  we can push the part of  $\Gamma$  on  $S_p$  outside it, except for those edges touching  $v_p$ .

Now, a solution of  $(G, H, \Pi, \mathcal{C})$  can be obtained by the following procedure, for each of the singular points  $p$ : (1) Remove the sphere replacing  $p$  in  $\mathcal{S}$ , a supersurface of  $\mathcal{C}$ , together with the image of  $\Gamma$  inside it; (2) add back the regular neighborhood  $N_p$  of  $p$  in  $\mathcal{C}$  that was removed when constructing  $\mathcal{S}$ ; (3) reconnect to  $p$  the points on the image of  $\Gamma$  that lie on the boundary of  $N_p$ .  $\square$

We now deduce from the previous EEP the desired EEP on a surface without boundary.

**Lemma 6.3.3.** *Any instance of EEP-SING( $n, m, c$ ) on a possibly disconnected surface with boundary reduces to an instance of EEP-SURF( $n + O(m), O(m), O(c)$ ).*

*Proof.* Figure 6.3 illustrates the proof. Let  $(G, H, \Pi, \mathcal{S})$  be an instance of an EEP on a possibly disconnected surface with boundary. We first describe the construction of  $(G', H', \Pi', \mathcal{S}')$ , the EEP instance on a possibly disconnected surface without boundary.

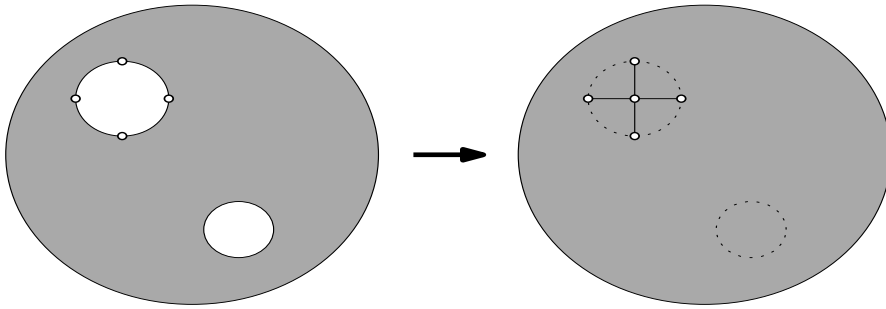


Figure 6.3: The removal of boundary components in the proof of Lemma [6.3.3](#). We attach a disk to every boundary component of the surface. Moreover, if there is a boundary component  $c$  containing at least one vertex of  $H$ , say  $v_1, \dots, v_k$ , we add to  $H$  a new vertex  $v$ , mapped inside the corresponding disk, and add, in  $H$ , one edge between  $v$  and each of the  $v_i$ s, that edge being mapped inside the disk.

Let  $\mathcal{S}'$  be obtained from  $\mathcal{S}$  by gluing a disk  $D_b$  along each boundary component. Let  $b$  be a boundary component of  $\mathcal{S}$ . If  $\Pi$  maps at least one vertex to  $b$ , then we add to  $H$  (and thus also to  $G$ ) a new vertex  $v_b$ , which we connect, also by a new edge, to each of the vertices mapped to  $b$  by  $\Pi$ . We extend  $\Pi$  by mapping vertex  $v_b$  and its incident edges inside  $D_b$ . Let us call  $G'$  and  $H'$  the resulting graphs, and  $\Pi'$  the resulting embedding of  $H'$ . For each vertex of  $H$  on a boundary component, we added to  $H$  and  $G$  at most one vertex and one edge and thus at most two branches. There remains to prove that these two EEPs are equivalent. Clearly, any solution of  $(G, H, \Pi, \mathcal{S})$  yields a solution of  $(G', H', \Pi', \mathcal{S}')$ . Conversely, let  $\Gamma'$  be a solution of  $(G', H', \Pi', \mathcal{S}')$ ; we build a solution of  $(G, H, \Pi, \mathcal{S})$ . Let  $b$  be a boundary of  $\mathcal{S}$ . There are two cases:

- If no vertex of  $H$  is mapped to  $b$ , then  $\Pi'$  maps  $H'$  outside the closure of  $D_b$ , so, by an ambient isotopy of  $\Gamma'$  that does not move  $H'$ , we can push  $\Gamma'$  outside  $D_b$ .
- Otherwise, the disk  $D_b$  is split into sectors by the edges incident to  $v_b$ ; the image of  $H'$  by  $\Pi'$  does not enter the interior of each sector, and intersects the boundary of a sector exactly along the image of  $v_b$  and of two edges incident to  $v_b$ . Thus, by an ambient isotopy that keeps the image of  $H'$  fixed, we can push the image of  $G'$  out of each sector.

After doing this for every boundary component  $b$ , we obtain that the restriction of  $\Gamma'$  to  $G$  is a solution of  $(G, H, \Pi, \mathcal{S})$ .  $\square$

Finally:

*Proof of Proposition [6.3.1](#).* It suffices to apply Lemmas [6.3.2](#) and [6.3.3](#).  $\square$



## 6.4 Reduction to a cellular EEP on a surface

Let **EEP-CELL** be the problem EEP, restricted to instances  $(G, H, \Pi, \mathcal{S})$  where  $\mathcal{S}$  is a surface and  $H$  is cellularly embedded and intersects each connected component of  $G$ .

In this section, we prove the following proposition.

**Proposition 6.4.1.** *Any instance of EEP-SURF( $n, m, c$ ) reduces to  $(n + m + c)^{O(m+c)}$  instances of EEP-CELL( $O(n + m + c), O(m + c), c$ ).*

For the proof of Proposition 6.4.1, which is given in the rest of Section 6.4, it will be necessary not to store an embedding  $\Pi$  of a graph  $G$  on a surface  $\mathcal{S}$  by its overlay with the triangulation, as was done in the previous section. Instead, we forget the triangulation. In other words, we have to store the combinatorial map corresponding to  $\Pi$ , but taking into account the fact that  $\Pi$  is not necessarily cellular. We need to store, for each face of the embedding, whether it is orientable or not, and a pointer to an edge of each of its boundary components (with some orientation information). Such a data structure is known under the name of *extended combinatorial map* [CM14, Section 2.2] (only orientable surfaces were considered there, but the data structure readily extends to non-orientable surfaces).

### 6.4.1 Reduction to connected surfaces

We first build intermediary EEPs over connected surfaces. Let **EEP-CONN** be the problem EEP, restricted to instances  $(G, H, \Pi, \mathcal{S})$  where  $\mathcal{S}$  is a surface (connected and without boundary) and  $H$  intersects every connected component of  $G$ .

**Lemma 6.4.2.** *Any instance of EEP-SURF( $n, m, c$ ) reduces to  $O(c(m + c)^c)$  instances of EEP-CONN( $n, m + c, c$ ).*

More precisely (and this is a fact that will be useful to prove that **EMBED** is in NP), any instance of EEP-SURF( $n, m, c$ ) is equivalent to the disjunction (OR) of at most  $(m + c)^c$  instances, each of them being the conjunction (AND) of  $O(c)$  instances of EEP-CONN( $n, m + c, c$ ).

*Proof.* We start by removing the connected components of  $G$  that are planar and disjoint from  $H$ . (Testing planarity takes linear time [HT74].) This does not change the solution of the EEP, because such connected components can be embedded on an arbitrarily small planar portion of  $\mathcal{S}$  (provided  $\mathcal{S}$  is non-empty, but otherwise the original EEP-SURF instance can be solved trivially). So without loss of generality, every connected component of  $G$  disjoint from  $H$  is non-planar. Let  $V_0$  be an arbitrary set of vertices, one per non-planar connected component of  $G$  disjoint from  $H$ . Without loss of generality, the number of vertices in  $V_0$  is at most the genus of  $\mathcal{S}$ , and hence at most  $c$ , because otherwise the initial EEP has no solution.

Let  $H' := H \cup V_0$ ; every connected component of  $G$  intersects  $H'$ . For each vertex of  $V_0$ , we guess the face of  $\Pi$  it has to be embedded in, and extend  $\Pi$  accordingly, by adding the images of  $V_0$  in  $\Pi$ ; let  $\Pi'$  be the resulting embedding of  $H'$ . By the previous

paragraph, the number of these guesses is at most  $(m + c)^{|V_0|} \leq (m + c)^c$ . It is clear that the initial EEP has a solution if and only if one of these EEPs  $(G, H', \Pi', \mathcal{S})$  has a solution.

These EEPs are almost of the form announced in the lemma, except that  $\mathcal{S}$  can be disconnected. However, in any solution of this EEP, we know the connected component of  $\mathcal{S}$  each connected component of  $G$  has to embed in, because each connected component of  $G$  intersects  $H$ . We can thus reformulate the EEP as the conjunction of several EEPs, one per connected component of  $\mathcal{S}$ . (Of course, we can discard the connected components of  $\mathcal{S}$  disjoint from  $H$ .)  $\square$

### 6.4.2 Simplifying the faces

The strategy for the proof of Proposition [6.4.1](#) is as follows. For each EEP  $(G', H', \Pi', \mathcal{S}')$  from the previous lemma, we will extend  $H'$  to make it cellularly embedded in  $\mathcal{S}'$  by adding either paths connecting two boundary components of a face of  $H'$ , or paths with endpoints on the same boundary component of a face of  $H'$  in a way that the genus of the face decreases. We first define an invariant that will allow us to control the number of steps needed until this process terminates.

Let  $\Pi$  be an embedding of a graph  $H$  on a surface  $\mathcal{S}$ . The *cellularity defect* of  $(H, \Pi, \mathcal{S})$  is the non-negative integer

$$cd(H, \Pi, \mathcal{S}) := \sum_{f \in \mathcal{F}(\Pi)} \text{genus}(f) + \sum_{f \in \mathcal{F}(\Pi)} (\text{number of boundaries of } f - 1)$$

where  $\mathcal{F}(\Pi)$  denotes the set of faces of  $\Pi$ .

Some obvious remarks:  $\Pi$  can contain isolated vertices. By convention, each of them counts as a boundary component of the face of  $\Pi$  it lies in. With this convention, every face of  $H$  has at least one boundary component, except in the very trivial case when  $G$  is empty. This implies that  $\Pi$  is a cellular embedding if and only if  $cd(H, \Pi, \mathcal{S}) = 0$ . We will also use the following property, usually without mentioning it.

**Lemma 6.4.3.** *Let  $\mathcal{S}$  be a 2-complex with  $c$  simplices that is homeomorphic to a surface and let  $H$  be a graph with  $m$  branches that is embedded in  $\mathcal{S}$ . Then  $cd(H, \Pi, \mathcal{S}) < 2m + c$ .*

*Proof.* Note that the genus of  $\mathcal{S}$  is smaller than  $c$  and hence

$$\sum_{f \in \mathcal{F}(\Pi)} \text{genus}(f) \leq \text{genus}(\mathcal{S}) < c.$$

Similarly, every boundary component contains a branch of  $H$  and each branch of  $H$  participates in at most two boundary components (or twice in the same boundary component). Thus,

$$\sum_{f \in \mathcal{F}(\Pi)} (\text{number of boundaries of } f - 1) \leq 2m.$$

$\square$

A *lollipop* in a graph is a closed path of the form  $p \cdot q \cdot \bar{p}$ , where  $p$  is a simple path (possibly reduced to a single vertex) and  $q$  is a cycle, such that  $p$  and  $q$  share exactly one vertex (the end-vertex of  $p$ ). The *basepoint* of the lollipop is the initial vertex of  $p$ . The following lemma reduces an EEP to EEPs with a smaller cellularity defect.

**Lemma 6.4.4.** *Any instance  $(G, H, \Pi, \mathcal{S})$  of  $\text{EEP-CONN}(n, m, c)$  with positive cellularity defect reduces to  $O(m^2 n^2)$  instances  $(G', H', \Pi', \mathcal{S})$  of  $\text{EEP-CONN}(n+1, m+3, c)$  where  $cd(H', \Pi', \mathcal{S}) < cd(H, \Pi, \mathcal{S})$ . For each such instance, the graph  $G'$  is obtained from  $G$  by adding exactly one edge, and the subgraph  $H'$  is obtained from  $H$  by adding a simple path or a lollipop containing the added edge that is embedded under  $\Pi'$  in one of the faces of the embedding  $\Pi$  of  $H$  in such a way that it does not separate that face.*

Admitting Lemma 6.4.4, the proof of Proposition 6.4.1 is straightforward:

*Proof of Proposition 6.4.1.* We first apply Lemma 6.4.2, obtaining  $O(c(m+c)^c)$  instances of  $\text{EEP-CONN}(n, m+c, c)$ . To each of these EEPs, we apply recursively Lemma 6.4.4 until we obtain cellular EEPs. The cellularity defect of the initial instance  $(G, H, \Pi, \mathcal{S})$  is  $O(m+c)$ , being at most the genus of  $\mathcal{S}$  plus  $2m$ , because each boundary component of a face of  $\Pi$  is incident to at least one branch of  $H$  (and each branch accounts for at most two boundary components in this way). Thus, the number of instances of  $\text{EEP-CELL}$  at the bottom of the recursion tree is  $(n+m+c)^{O(m+c)}$ , in each of which the size of the graph is  $O(n+m+c)$ , the pre-embedded subgraph has  $O(m+c)$  branches, and the surface is the same, thus having at most  $c$  simplices.  $\square$

### 6.4.3 Proof of Lemma 6.4.4

It remains to prove Lemma 6.4.4. The proof uses standard notions in surface topology, homotopy, and homology. We refer to textbooks and surveys [MT01; Sti93; Col18]. We only consider homology with coefficients in  $\mathbb{Z}/2\mathbb{Z}$ .

We start with some auxiliary definitions and lemmas. The reversal of a path  $w$  of a graph is denoted by  $\bar{w}$ . The concatenation of two paths  $w$  and  $w'$  is denoted by  $w \cdot w'$ . We use the same notation for paths on a surface.

Let  $f$  be a surface with a single boundary component and let  $a$  be a (possibly non-simple) arc in  $f$  intersecting the boundary component of  $f$  exactly at its endpoints. If we contract the boundary component to a single point, the arc  $a$  becomes a closed simple curve, which can be null-homologous or non-null-homologous, and one-sided or two-sided. We employ the same adjectives, null-homologous, non-null-homologous, one-sided, and two-sided, for the arc  $a$ .

**Lemma 6.4.5.** *Let  $f$  be a surface with boundary, let  $b$  be a point in the interior of  $f$ , and let  $b_1, b_2,$  and  $b_3$  be points on the boundary of  $f$ . For each  $i$ , let  $p_i$  be a (possibly non-simple) path connecting  $b_i$  to  $b$ . Let  $a_1 = p_2 \cdot \bar{p}_3$ ,  $a_2 = p_3 \cdot \bar{p}_1$ , and  $a_3 = p_1 \cdot \bar{p}_2$ . Let  $\mathcal{P}$  be any of the following possible properties of the arcs  $a_i$ :*

- “the endpoints of  $a_i$  lie on the same boundary component of  $f$ ”;

- “ $a_i$  is null-homologous” (if  $f$  has a single boundary component);
- “ $a_i$  is two-sided” (if  $f$  has a single boundary component).

Then the following holds: If both  $a_1$  and  $a_2$  have property  $\mathcal{P}$ , then so has  $a_3$ .

*Proof.* This is a variant on the 3-path condition from Mohar and Thomassen [MT01, Section 4.3]. The first item is immediate. The second one follows from the fact that homology is an algebraic condition: The concatenation of two null-homologous paths is null-homologous, and removing spurs (subpaths of the form  $q \cdot \bar{q}$ ) from a path does not affect homology. The third one is similar: The concatenation of two two-sided paths is two-sided, and removing spurs does not affect sidedness.  $\square$

**Lemma 6.4.6.** *Let  $G$  be a graph,  $H$  a subgraph of  $G$ , and  $\Pi$  be an embedding of  $H$  on  $\mathcal{S}$ . Let  $K$  be a subgraph of  $G$  that is either a simple path intersecting  $H$  exactly at its endpoints, or a lollipop intersecting  $H$  exactly at its basepoint. Let  $\Pi'$  be an embedding of  $H \cup K$  extending  $\Pi$  such that the embedding of  $K$  does not separate the face  $f$  of  $\Pi$  it belongs to. Let  $f'$  be the face of  $\Pi'$  corresponding to  $f$ . Then  $cd(H \cup K, \Pi', \mathcal{S}) = cd(H, \Pi, \mathcal{S}) - 1$ . Moreover:*

1. *If  $K$  has its endpoints on two distinct boundary components of  $f$ , then  $f'$  has the same genus and orientability character as  $f$ , and one boundary component less than  $f$ .*
2. *If  $K$  has its endpoints on the same boundary component of  $f$ , then we have the following possibilities:*
  - *if  $K$  is two-sided, then  $f'$  has the same orientability characteristic as  $f$ ; moreover, the genus of  $f'$  equals that of  $f$ , minus two, and the number of boundary components of  $f'$  equals that of  $f$ , plus one;*
  - *if  $K$  is one-sided, then  $f$  is non-orientable, the genus of  $f'$  equals that of  $f$  minus one, and the number of boundary components of  $f'$  equals that of  $f$ .*

*Moreover:*

    - *if the genus of  $f$  is one, then  $f'$  is orientable;*
    - *if the genus of  $f$  is odd and at least 3, then  $f'$  may be either orientable or non-orientable;*
    - *if the genus of  $f$  is even, then  $f'$  is non-orientable.*

*Proof.* Let us first prove that  $cd(H \cup K, \Pi', \mathcal{S}) = cd(H, \Pi, \mathcal{S}) - 1$ . By the choice of  $K$ , we have that the Euler characteristic of  $f'$  exceeds that of  $f$  exactly by one. Moreover, the Euler characteristic of  $f$  equals two, minus its genus, minus its number of boundary components, and similarly for  $f'$ . The result on the cellularity defect follows.

The other assertions also follow from Euler characteristic arguments. Specifically, we first remark that, in all cases, the claim on the number of boundary components of  $f'$  is correct (this only depends on whether  $K$  connects the same or different boundary

components of  $f$ , and whether it is one- or two-sided). Then the genus follows, as above, from Euler characteristic arguments. The claims on the orientability use the following facts: If  $f$  is orientable, then so is  $f'$ ; if  $f'$  has genus zero, then it is orientable; if  $f'$  has odd genus, then it is non-orientable.  $\square$

*Proof of Lemma 6.4.4.* Since  $cd(H, \Pi, \mathcal{S}) \geq 1$ , there must be a face  $f$  of  $H$  with either (1) at least two boundary components, or (2) a single boundary component but positive genus. We will consider each of these cases separately, but first we introduce some common terminology.

Let  $F$  be an arbitrary spanning forest of  $G - E(H)$  rooted at  $V(H)$ . This means that  $F$  is a subgraph of  $G - E(H)$  that is a forest with vertex set  $V(G)$  such that each connected component of  $F$  contains exactly one vertex of  $V(H)$ , its *root*, and each vertex of  $G$  is in  $F$ . The algorithm starts by computing an arbitrary such forest  $F$  in linear time.

For each vertex  $u$  of  $G$ , let  $r(u)$  be the unique root in the same connected component of  $F$  as  $u$ , and let  $F(u)$  be the unique path connecting  $u$  to  $r(u)$ . If  $u$  and  $v$  are two vertices of  $G$ , let  $G_{uv}$  be the graph obtained from  $G$  by adding one edge, denoted  $uv$ , connecting  $u$  and  $v$ . (This may be a parallel edge if  $u$  and  $v$  were already adjacent in  $G$ , but in such a situation when we talk about edge  $uv$  we always mean the new edge.) Let  $F(uv)$  be the unique path in  $G$  between  $u$  and  $v$  that is the concatenation of  $\overline{F(u)}$ , edge  $uv$ , and  $F(v)$ . Thus,  $F(uv)$  intersects  $H$  precisely at its endpoints. Note that  $F(uv)$  is either a simple path with its ends in  $H$  or a lollipop with its basepoint in  $H$  (and otherwise disjoint from  $H$ ).

**Case 1:  $f$  has at least two boundary components.**

Assume that  $(G, H, \Pi, \mathcal{S})$  has a solution  $\Gamma$ . We claim that, *for some vertices  $u$  and  $v$  of  $G$ , the embedding  $\Gamma$  extends to an embedding of  $G_{uv}$  in which the image of  $F(uv)$  lies in  $f$  and connects two distinct boundary components of  $f$ .*

Indeed, let  $\gamma$  be a curve drawn in  $f$  connecting two vertices of  $H$  in different boundary components of  $f$  and chosen such that it intersects the boundary of  $f$  exactly at its endpoints. We can deform  $\gamma$  so that it intersects  $\Gamma$  only at the images of vertices, and never in the relative interior of an edge. We can, moreover, assume that  $\gamma$  is simple (except perhaps that its endpoints may coincide if they join occurrences of the same vertex in different boundary components of  $f$ ). Let  $v_1, \dots, v_k$  be the vertices of  $G$  encountered by  $\gamma$ , in this order. We denote by  $\gamma[i, j]$  the segment of  $\gamma$  between vertices  $v_i$  and  $v_j$ . For some  $i$ , we have that  $\overline{F(v_i)} \cdot \gamma[i, i+1] \cdot F(v_{i+1})$  connects two different boundary components of  $f$ : Otherwise, by induction on  $i$ , applying the first case of Lemma 6.4.5 to the three paths  $\gamma[1, i]$ ,  $F(v_i)$ , and  $\gamma[i, i+1] \cdot F(v_{i+1})$ , we would have that, for each  $i$ ,  $\gamma[1, i] \cdot F(v_i)$  has its endpoints on the same boundary component of  $f$ , which is a contradiction for  $i = k$  (for which the curve is  $\gamma$ ). So let  $i$  be such that  $\overline{F(v_i)} \cdot \gamma[i, i+1] \cdot F(v_{i+1})$  connects two different boundary components of  $f$ . By letting  $u = v_i$  and  $v = v_{i+1}$ , and embedding the edge  $uv$  as  $\gamma[i, i+1]$ , gives the desired embedding of  $G_{uv}$ . This proves the claim.

By definition of  $F_{uv}$ , and since it connects two distinct boundary components of  $f$ , it is actually a path (without repeated vertices) in  $f$ . (Its endpoints may coincide on  $\mathcal{S}$ .)

The strategy now is to guess the vertices  $u$  and  $v$  and the way the path  $F(uv)$  is drawn in  $f$ , and to solve a set of EEPs  $(G_{uv}, H \cup F(uv), \Pi', \mathcal{S})$  where  $\Pi'$  is chosen as an appropriate extension of  $\Pi$ .

*Let us first assume that  $f$  is orientable.* One subtlety is that, given  $u$  and  $v$ , there can be several essentially different ways of embedding  $F(uv)$  inside  $f$ , if there is more than one occurrence of  $r(u)$  and  $r(v)$  on the boundary of  $f$ . So we reduce our EEP to the following set of EEPs: For each choice of vertices  $u$  and  $v$  of  $G$ , and any occurrences of  $r(u)$  and  $r(v)$  on the boundary of  $f$ , we consider the EEP  $(G_{uv}, H \cup F(uv), \Pi', \mathcal{S})$  where  $\Pi'$  extends  $\Pi$  and maps  $F(uv)$  to an arbitrary path in  $f$  connecting the chosen occurrences of  $r(u)$  and  $r(v)$  on the boundary of  $f$ .

It is clear that, if one of these new EEPs has a solution, the original EEP has a solution. Conversely, let us assume that the original EEP  $(G, H, \Pi, \mathcal{S})$  has a solution; we now prove that one of these new EEPs has a solution. By our claim above, for some choice of  $u$  and  $v$ , some EEP  $(G_{uv}, H \cup F(uv), \Pi'', \mathcal{S})$  has a solution, for some  $\Pi''$  mapping  $F(uv)$  inside  $f$  and connecting different boundary components of  $f$ . In that mapping,  $F(uv)$  connects two occurrences of  $r(u)$  and  $r(v)$  inside  $f$ . We prove that, for these choices of occurrences of  $r(u)$  and  $r(v)$ , the corresponding EEP described in the previous paragraph,  $(G_{uv}, H \cup F(uv), \Pi', \mathcal{S})$ , has a solution as well. These two EEPs are the same except that the path  $F(uv)$  may be drawn differently in  $\Pi'$  and  $\Pi''$ , although they connect the same occurrences of  $r(u)$  and  $r(v)$  on the boundary of  $f$ . By Lemma [6.4.6](#), under  $\Pi'$ , the face  $f$  is transformed into a face  $f'$  that has the same genus and orientability character as  $f$ , but one boundary component less. The same holds, of course, for  $\Pi''$ . Moreover, the ordering of the vertices on the boundary components of the new face is the same in  $\Pi'$  and  $\Pi''$ . Thus, there is a homeomorphism  $h$  of  $f$  that keeps the boundary of  $f$  fixed pointwise and such that  $h \circ \Pi''|_{F(uv)} = \Pi'|_{F(uv)}$ . This homeomorphism, extended to the identity outside  $f$ , maps any solution of  $(G_{uv}, H \cup F(uv), \Pi'', \mathcal{S})$  to a solution of  $(G_{uv}, H \cup F(uv), \Pi', \mathcal{S})$ , as desired.

It also follows from Lemma [6.4.6](#) that the cellularity defect decreases by one. To conclude this case, we note that the number of new EEPs is  $O(m^2n^2)$ : indeed, there are  $O(n)$  possibilities for the choice of each of  $u$  and  $v$ , and  $O(m)$  possibilities for the choice of each of the occurrence of  $r(u)$  and  $r(v)$  on the boundary of  $f$ .

*If  $f$  is non-orientable,* the same argument works, except that there are two possibilities for the cyclic ordering of the vertices along the new boundary component of the new face: If we walk along one of the boundary components of  $f$  (in an arbitrary direction), use  $p$ , and walk along the other boundary component of  $f$ , we do not know in which direction this second boundary component is visited. So we actually need to consider two EEPs for each choice of  $u$ ,  $v$ , and occurrences of  $r(u)$  and  $r(v)$ , instead of one. The rest is unchanged.

**Case 2:  $f$  has a single boundary component and positive genus.** The proof is similar to Case 1. The main difference is that, instead of curves connecting different boundary components of  $f$ , we now consider curves in  $f$  that are non-null-homologous. Another difference is that the walks  $F(uv)$  may repeat vertices and edges; however, by construction,  $F_{uv}$  is either a (simple) path or a lollipop.

Assume that  $(G, H, \Pi, \mathcal{S})$  has a solution  $\Gamma$ . We claim that, for some vertices  $u$  and  $v$  of  $G$  (allowing the possibility that  $u = v$ ), the embedding  $\Gamma$  extends to an embedding of  $G_{uv}$  in which the image of  $F(uv)$  lies in  $f$  and is non-null-homologous. The proof is similar in spirit to the corresponding claim in Case 1: We let  $\gamma$  be a non-null-homologous curve in  $f$  intersecting the boundary of  $f$  exactly at its endpoints; we can assume similarly as before that it is simple and intersects only vertices of  $\Gamma$ , in the order  $v_1, \dots, v_k$ . For some  $i$ ,  $\overline{F(v_i)} \cdot \gamma[i, i+1] \cdot F[v_{i+1}]$  must be non-null-homologous, by induction and by Lemma 6.4.5; this gives an embedding of  $G_{uv}$ . Moreover, if  $f$  is non-orientable,  $\gamma$  can be chosen to be one-sided. Then the above proof also shows that the image of  $F(uv)$  can be chosen so that it will be one-sided.

If  $f$  is orientable, we reduce the original EEP to the following set of EEPs: For each choice of vertices  $u$  and  $v$  of  $G$ , and each occurrence of  $r(u)$  and  $r(v)$  on the boundary of  $f$ , we consider the EEP  $(G_{uv}, H \cup F(uv), \Pi', \mathcal{S})$  where  $\Pi'$  extends  $\Pi$  and embeds  $F(uv)$  in  $f$  in such a way that (1)  $F(uv)$  is non-null-homologous, and (2)  $F(uv)$  connects the chosen occurrences of  $r(u)$  and  $r(v)$  on the boundary of  $f$ . As before, the only subtlety is to prove that, if we have two EEPs  $(G_{uv}, H \cup F(uv), \Pi', \mathcal{S})$  and  $(G_{uv}, H \cup F(uv), \Pi'', \mathcal{S})$  such that  $F(uv)$  are not embedded exactly in the same way in  $\Pi'$  and  $\Pi''$ , but are non-null-homologous in  $f$  and connect the same occurrences of  $r(u)$  and  $r(v)$  on the boundary of  $f$ , then these EEPs are equivalent. This follows from Lemma 6.4.6: The image of  $F(uv)$  in  $\Pi'$  is non-null-homologous, thus non-separating, and thus cuts  $f$  into a face that is an orientable surface with two boundary components and with (Euler) genus that of  $f$  minus two; the same holds in  $\Pi''$ . Moreover, the ordering of the vertices along the boundary components of the new face is the same in both  $\Pi'$  and  $\Pi''$ . Thus, as in the previously treated case, there is a homeomorphism  $h$  of  $f$  that keeps the boundary of  $f$  fixed pointwise and such that  $h \circ \Pi''|_{F(uv)} = \Pi'|_{F(uv)}$ . We complete the proof in the same way as before. It also follows from Lemma 6.4.6 that the cellularity defect decreases by one. The number of these new EEPs is  $O(m^2 n^2)$ .

If  $f$  is non-orientable, then we consider only embeddings where the embedding of  $F(uv)$  in  $f$  is one-sided (which is possible by Lemma 6.4.5). If the genus of  $f$  is even or equal to 1, then a similar argument can be used: Regardless of the way we embed  $F(uv)$  in a way that (1) it is one-sided and (2) it connects the chosen occurrences  $r(u)$  and  $r(v)$  on the boundary of  $f$ , cutting  $f$  along  $F(uv)$  results in a surface whose topology is uniquely determined, by Lemma 6.4.6. (Note that  $F(uv)$  is non-separating because it is one-sided.) Moreover, the ordering of the vertices along the boundary of the new face is uniquely determined. The cellularity defect also decreases by one, and the same argument as above concludes.

Finally, if the genus of  $f$  is odd and at least three, then cutting  $f$  along  $F(uv)$  results in a surface in which the ordering of the vertices along the single boundary component is uniquely determined, but this surface, with one boundary component and with genus that of  $f$  minus one, can be orientable or not. Thus, for each choice of vertices  $u$  and  $v$  of  $G$ , and each occurrence of  $r(u)$  and  $r(v)$  on the boundary of  $f$ , we actually need to consider two EEPs, one in which  $F(uv)$  is embedded as a one-sided curve in a way that it cuts  $f$  into an orientable surface, and one in which  $F(uv)$  is embedded in a way that

it cuts  $f$  into a non-orientable surface. The rest of the argument is unchanged.  $\square$

## 6.5 Solving a cellular EEP on a surface

**Theorem 6.5.1.** *There is a function  $f : \mathbb{N}^2 \rightarrow \mathbb{N}$  such that every instance of EEP-CELL( $n, m, c$ ) can be solved in time  $f(m, c) \cdot O(n)$ .*

When solving EEPs on surfaces, a useful property of the embedded subgraph is the following one. We say that a subgraph  $H$  of  $G$  has *property (E)* if  $H$  has no local bridges. This means that every branch of  $H$  forms an induced subgraph of  $G$ , and for every connected component of  $G - V(H)$ , its neighbors in  $H$  are not all contained in a single branch of  $H$ . In fact, what we need is a weaker version of property (E) where we first prescribe a subset  $V_0$  of vertices of  $H$ , where  $V_0$  contains all vertices whose degree in  $H$  is different from two, and possibly a constant number of vertices whose degree in  $H$  is equal to two. To each vertex in  $V_0$ , whose degree is 2 in  $H$ , we add an edge to make it a degree-3 vertex. Let  $\hat{H}$  be the resulting graph. Then we say that  $H$  has property (E) with respect to  $V_0$  if the above property holds with respect to the branches of  $\hat{H}$ .

*Proof of Theorem 6.5.1.* This is essentially the main result from [Moh99]. The algorithm from [Moh99] first reduces the problem to an instance  $(G', H', \Pi', \mathcal{S})$  such that  $H'$  satisfies property (E) with respect to a subset  $V_0$  that contains all vertices of  $H$  of degree different from two (all these are also vertices in  $H'$  of degree different from two) plus a constant number of vertices of degree two in  $H'$ , where this constant number is bounded from above in terms of the genus of  $\mathcal{S}$ , which is itself bounded from above by  $c$ . For this purpose, [Moh99] relies on another paper [JMM97]. After achieving this property, the paper [Moh99] reduces the EEP to a constant number (where the constant depends on  $c$ ) of “simple” extension problems [Moh99, Section 4], which are then solved in [Moh99, Theorem 5.4].  $\square$

## 6.6 Proof of Theorems 6.1.1 and 6.1.2

We can finally prove our main results. First, let us prove that we have an algorithm with complexity  $f_0(c) \cdot n^{O(c)}$ .

*Proof of Theorem 6.1.2.* This immediately follows from Propositions 5.2.1, 6.2.1, 6.3.1, 6.4.1, and Theorem 6.5.1. Consider an instance of EMBED( $n, c$ ). Proposition 5.2.1 allows to discard the 2-complexes containing a 3-book. Proposition 6.2.1 reduces the problem to  $(cn)^{O(c)}$  instances of EEP-SING( $cn, c, O(c)$ ). Proposition 6.3.1 reduces each such instance into an instance of EEP-SURF( $O(cn), O(c), O(c)$ ). Proposition 6.4.1 reduces that instance into  $O(cn)^{O(c)}$  instances of EEP-CELL( $O(cn), O(c), O(c)$ ). Theorem 6.5.1 shows that each such instance can be solved in time  $f(O(c), O(c)) \cdot O(cn) \leq f_1(c)n$  for appropriate function  $f_1 : \mathbb{N} \rightarrow \mathbb{N}$ .  $\square$

Finally, we prove that the problem EMBED is NP-complete:



*Proof of NP-completeness.* Let us first prove that the problem is NP-hard. The following problem GRAPH-GENUS is NP-hard: Given a graph  $G$  and an integer  $g$ , decide whether  $G$  embeds on the orientable surface of genus  $g$  [Tho89]. This almost immediately implies that EMBED is NP-hard; the only subtlety is that in GRAPH-GENUS,  $g$  is specified in binary, thus more compactly than a triangulated surface of genus  $g$  (and thus  $\Omega(g)$  triangles). To be very precise, given an instance  $(G, g)$  of GRAPH-GENUS, we transform it in polynomial time into an equivalent instance of EMBED as follows: If  $G$  has at most  $g$  edges, then we transform it into a constant-size positive instance of EMBED (every graph with  $g$  edges embeds on the orientable surface of genus  $g$ ); otherwise, we consider the instance  $(G, \mathcal{C})$  where  $\mathcal{C}$  is a 2-complex that is an orientable surface of genus  $g$ ; since  $G$  has at least  $g$  edges, the transformation takes polynomial time in the size of  $(G, g)$ .

We now prove that the problem EMBED belongs to NP. The case where  $\mathcal{C}$  contains a 3-book is trivial; let us assume that it is not the case. The proof of Proposition 6.3.1 shows that an EMBED instance is positive if and only if at least one instance of EEP-SURF, among  $(cn)^{O(c)}$  of them, is positive. The certificate indicates which of these instances is positive (this requires a polynomial number of bits), together with a certificate that this instance is indeed positive (see below). To check this certificate, the algorithm builds the corresponding instance of EEP-SURF (as done in Section 6.3—this takes polynomial time) and checks the certificate.

Here is a way to provide a certificate for an instance of EEP-SURF. In Section 6.4 we have proved that, if we have an instance  $(G, H, \Pi, \mathcal{S})$  of EEP-SURF, then there exists a cellular embedding  $\Gamma'$  (in the form of a combinatorial map) of a graph  $G'$  containing  $G$ , and such that  $\Gamma'$  extends  $\Pi$ . Moreover,  $G'$  is obtained from  $G$  by adding a number of edges that is  $O(c)$ , where  $c$  is the size of the original complex. (Recall that in the instance of EEP-SURF, the size of  $H$  is  $O(c)$ .) The cellular embedding  $\Gamma'$  of  $G'$ , given as a combinatorial map, is the certificate that  $(G, H, \Pi, \mathcal{S})$  is positive: Given  $(G, H, \Pi, \mathcal{S})$  and this certificate, we can in polynomial time check that  $G'$  contains  $G$ , that the restriction of  $\Gamma'$  to  $H$  is indeed  $\Pi$ , and that the combinatorial map of  $\Gamma'$  is indeed an embedding on  $\mathcal{S}$ .  $\square$

# Embedding graphs on 2-complexes: a fixed-parameter tractable algorithm

The contents of this chapter, written jointly with Éric Colin de Verdière will appear in 29th Annual European Symposium on Algorithms (ESA 2021).

## 7.1 Overview and structure of the chapter

In this chapter we study the *EMBED* problem: given a graph  $G$  and a 2-complex  $\mathcal{C}$ , decide whether there exists an embedding of  $G$  on  $\mathcal{C}$ . We describe algorithms that are fixed-parameter tractable in the complexity of the input complex deciding it.

**Theorem 7.1.1.** *One can solve the embeddability problem of graphs into 2-dimensional simplicial complexes in deterministic  $f(c)n^3$  or in expected time  $2^{c^{O(1)}}n^{O(1)}$ , where  $c$  is the number of simplices of the input 2-complex,  $n$  is the number of vertices and edges of the input graph and  $f$  is some computable function of  $c$ .*

Our strategy is standard in graph algorithms and parameterized complexity (see, e.g., the book by Cygan et al. [Cyg+15], Chapter 7): we show by dynamic programming that the problem can be solved efficiently for graphs of bounded branchwidth, and then, using an irrelevant vertex method, we prove that one can assume without loss of generality that the input graph  $G$  has branchwidth bounded by a polynomial in the size of the input 2-complex. In the context of surface-embedded graphs, this paradigm has been used in the extended abstract by Kawarabayashi et al. [KMR08] and in the article by Kociumaka and Pilipczuk [KP19]; our algorithm takes inspiration from the former, for the idea of the dynamic programming algorithm, and from the latter, for some arguments in the irrelevant vertex method. However, handling 2-complexes requires significantly more effort. More precisely, Theorem 7.1.1 follows immediately from the following two theorems.

**Theorem 7.1.2.** *One can solve the embeddability problem of graphs into two-dimensional simplicial complexes in time  $(c+w)^{O(c+w)}n$ , where  $c$  is the number of simplices of the input 2-complex, and where  $n$  and  $w$  are the number of vertices and edges and the branchwidth of the input graph, respectively.*

**Theorem 7.1.3.** *Let  $\mathcal{C}$  be a 2-complex with  $c$  simplices and  $G$  a graph with  $n$  vertices and edges. We can correctly report that  $G$  is embeddable on  $\mathcal{C}$ , or correctly report that  $G$  is not embeddable on  $\mathcal{C}$ , or compute a subgraph  $H$  of  $G$ , of branchwidth polynomial in the number of simplices of  $\mathcal{C}$ , such that  $G$  embeds on  $\mathcal{C}$  if and only if  $H$  does:*

- *in deterministic time  $f(c) \cdot n^3$ ,*
- *or in expected polynomial time.*

We now present the structure of the chapter, indicating which techniques are used. We also emphasize which components would be simpler if we were just aiming for an algorithm for embedding graphs on surfaces.

In Section 7.2, we show that we can make some simple assumptions on the input, and present data structures for representing 2-complexes and graphs embedded on them. If we restrict ourselves to the case where the input 2-complex is homeomorphic to a surface, we essentially consider combinatorial maps of graphs on surfaces, except that the graphs need not be cellularly embedded (such a data structure is called an *extended combinatorial map* [CM14, Section 2.2]). The case of 2-complexes is largely more involved.

In Section 7.3, we show that if our input graph  $G$  has an embedding into our input 2-complex  $\mathcal{C}$ , then there exists an embedding of  $G$  on  $\mathcal{C}$  that is *sparse* with respect to a branch decomposition of  $G$ . This means that each subgraph of  $G$  induced by the leaves of any subtree of the branch decomposition can be separated from the rest of  $G$  using a graph embedded on  $\mathcal{C}$ , called *partitioning graph*, of small complexity. We find that this new structural result, even in the surface case, is interesting and can prove useful in other contexts. If the target space were a surface, we could assume that  $G$  is 3-connected and has no loop or multiple edges, which would imply (still with some work) that *any* embedding of  $G$  would be sparse, but again the fact that we consider 2-complexes requires additional work.

In Section 7.4, we present the dynamic programming algorithm, which either determines the existence of an embedding of  $G$  on  $\mathcal{C}$ , or shows that no sparse embedding of  $G$  on  $\mathcal{C}$  exists (and thus no embedding at all, by the previous paragraph). The idea is to use bottom-up dynamic programming and to consider all regions of the 2-complex in which the subgraph of  $G$  (induced by a subtree of the branch decomposition) can be embedded. The complexity depends exponentially on the branchwidth of  $G$ .

The previous arguments, most notably in Section 7.3, implicitly assumed that, if  $G$  has an embedding into  $\mathcal{C}$ , it has a *proper and cellular embedding*, in particular, in which the faces are homeomorphic to disks. In Section 7.5, we show that we can assume this property. Essentially, we build all 2-complexes “smaller” than  $\mathcal{C}$ , such that  $G$  embeds on  $\mathcal{C}$  if and only if it embeds into (at least) one of these 2-complexes, and moreover if it is the case, it has an embedding into (at least) one of these 2-complexes that is proper

and cellular. If  $\mathcal{C}$  were an orientable surface, we would just consider the surfaces of lower genus; but here a more sophisticated approach is needed.

The above ingredients allow to prove Theorem 7.1.2 (Section 7.6).

In Section 7.7, we show, using an irrelevant vertex method, that we can assume that  $G$  has branchwidth polynomial in the size of  $\mathcal{C}$  (Theorem 7.1.3). In a nutshell, if  $G$  has large branchwidth, then using the result by Chekuri and Chuzhoy [CC16], we can compute a subdivision of a large wall, and then (unless  $G$  has large genus and is not embeddable on  $\mathcal{C}$ ) compute a large planar part of  $G$  containing a large wall; the central vertex of this wall is irrelevant, in the sense that its removal does not affect the embeddability or non-embeddability of the graph into  $\mathcal{C}$ ; iterating, we obtain a graph of branchwidth polynomial in the size of  $\mathcal{C}$ .

## 7.2 Preprocessing and data structures on 2-complexes

### 7.2.1 Some preprocessing

**Proposition 7.2.1.** *To decide the embeddability of a graph  $G$  on a 2-complex  $\mathcal{C}$ , we can without loss of generality, after a linear-time preprocessing, assume the following properties on the input:*

- $\mathcal{C}$  has no 3-book and no connected component that is reduced to a single vertex;
- $G$  has no connected component reduced to a single vertex, and at most one connected component homeomorphic to a segment.

*Proof.* It is known that every graph can be embedded into a 3-book (see Section 5.2.1). So we can without loss of generality assume that  $\mathcal{C}$  contains no 3-book. We remove all the isolated vertices of  $\mathcal{C}$ , and remove the same number of isolated vertices of  $G$  (to the extent possible); this does not affect whether  $G$  embeds into  $\mathcal{C}$ . We then replace each isolated vertex of  $G$  with an isolated edge; since  $\mathcal{C}$  has no more isolated vertex, this does not affect embeddability of  $G$  into  $\mathcal{C}$ . Finally, for the same reason, if  $G$  contains at least two connected components homeomorphic to segments, we replace all these components with a single edge.  $\square$

In the rest of this chapter, without loss of generality, we implicitly assume that  $\mathcal{C}$  and  $G$  satisfy the properties stated in Proposition 7.2.1.

### 7.2.2 Detached surface

**Detaching** a singular point  $p$  in  $\mathcal{C}$  consists of the following operation: replace  $p$  with new vertices, one for each cone, corner, and isolated segment at  $p$ . Detaching all singular points of a 2-complex without isolated vertices yields a disjoint union of (1) isolated segments and (2) a surface, possibly disconnected, possibly with boundary, called the **detached surface** (see Figure 7.1, right). The trace of the singular points on the detached surface are the **marked points**. Conversely,  $\mathcal{C}$  can be obtained from a surface

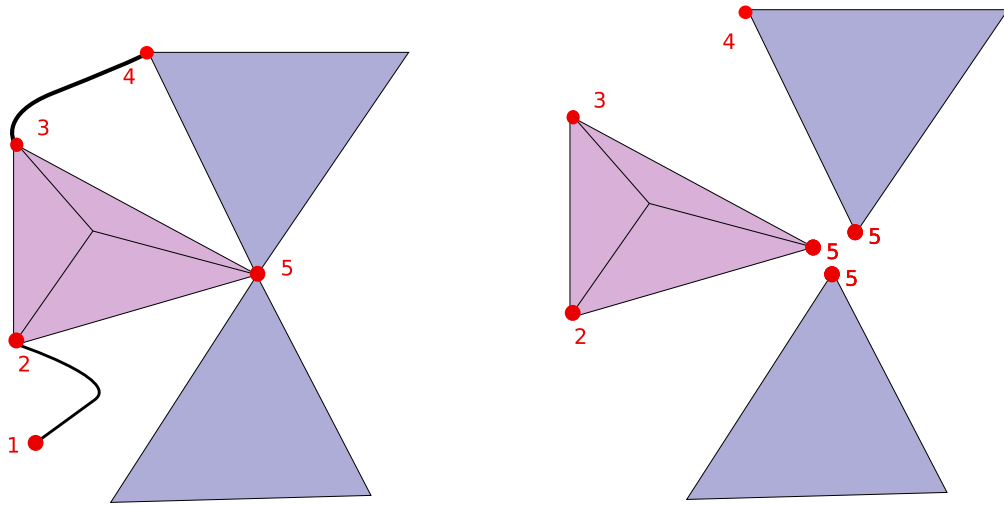


Figure 7.1: On the left: the complex  $\mathcal{C}$  of Figure 1.4. On the right: the corresponding detached surface.

(possibly disconnected, possibly with boundary) and a finite set of segments by choosing finitely many subsets of points and identifying the points in each subset together.

The *boundary* of  $\mathcal{C}$  is the closure of the set of points of  $\mathcal{C}$  that have an open neighborhood homeomorphic to a closed half-plane. Equivalently, it is the union of the edges of  $\mathcal{C}$  incident with a single triangle.

### 7.2.3 Topological data structure for 2-complexes

We now describe a *topological data structure* for 2-complexes without 3-book or isolated vertex that is more appropriate for our purposes. It records only the 2-complex up to homeomorphism, not the combinatorial information given by its simplices. Such a 2-complex  $\mathcal{C}$  is obtained from a surface  $\mathcal{S}$  (possibly disconnected, possibly with boundary) and a finite set  $S$  of segments by identifying together finitely many finite subsets of points. Our data structure stores separately the detached surface  $\mathcal{S}$ , the set  $S$  of isolated segments, and the singular points, and two-way pointers representing incidences between them. In more detail:

- we store the list of the connected components of the detached surface  $\mathcal{S}$ , and for each such component  $\mathcal{S}'$  we store (1) whether it is orientable or not; (2) its genus; (3) a list of pointers to the singular points in the interior of  $\mathcal{S}'$ ; (4) for each boundary component of  $\mathcal{S}'$ , a cyclically ordered list of pointers to the singular points appearing on that boundary component (if  $\mathcal{S}'$  is orientable, the boundary components must be traversed in an order consistent with an arbitrarily chosen orientation of  $\mathcal{S}'$ );
- we store the list  $S$  of isolated segments, and for each of them, two pointers to the singular points at its endpoints;

- conversely, to each singular point is attached a list of pointers to the occurrences of that singular point on the detached surface or as an endpoint of an isolated segment.

The *size* of a 2-complex (without 3-book or isolated vertex) is the sum of the number of isolated segments, the number of connected components of the detached surface, the total genus of the detached surface, the total number of boundary components of the detached surface, and the total number of marked points of the detached surface (the occurrences of the singular points). This is, up to a constant factor, the size of the topological data structure indicated above, if the genus is stored in unary.

Given a 2-complex  $\mathcal{C}$  without 3-book or isolated vertex, described by vertices, edges, and triangles and their incidence relations, we can easily compute a representation of  $\mathcal{C}$  in that data structure, in polynomial time: Indeed, by ignoring the incidences created by vertices, we easily build a triangulation of the surface  $\mathcal{S}$  (possibly disconnected, possibly with boundary) and a list of segments  $S$ ; we then compute the topology of  $\mathcal{S}$ ; finally, we mark the singular points, which are the vertices with several occurrences on  $\mathcal{S}$  and/or on  $S$ . We remark that the size of the resulting data structure is at most linear in the number of vertices, edges, and triangles of the 2-complex  $\mathcal{C}$ , because, by Euler's formula, any triangulated surface (possibly with boundary) with  $k$  simplices has genus  $O(k)$  and a number of boundary components that is  $O(k)$ . Thus, **in the rest of this chapter, without loss of generality, we implicitly assume that  $\mathcal{C}$  is given in the form of the above topological data structure.** (Conversely, it is not hard to see that every 2-complex is homeomorphic to a 2-complex whose number of simplices that is linear in its size, but we will not need this fact.)

We will need the following lemma.

**Lemma 7.2.2.** *Given two 2-complexes  $\mathcal{C}$  and  $\mathcal{C}'$ , given in the topological representation above, of sizes  $c$  and  $c'$  respectively, we can decide whether  $\mathcal{C}$  and  $\mathcal{C}'$  are homeomorphic in time  $(c + c')^{O(c+c')}$ .*

*Proof.* (We remark that this essentially follows from more general results [DWW00]; the running time of our algorithm might be improvable, but this suffices for our purposes.) As a preprocessing, in the topological data structures of  $\mathcal{C}$  and  $\mathcal{C}'$ , we do the following: whenever a singular point is incident to exactly two isolated segments and is not incident to the detached surface, we dissolve that singular point, removing it and replacing the two incident edges with a single one. Clearly, this does not affect whether  $\mathcal{C}$  and  $\mathcal{C}'$  are homeomorphic.

After this preprocessing,  $\mathcal{C}$  and  $\mathcal{C}'$  are homeomorphic if and only if their topological data structures are isomorphic. By this, we mean that there is a bijective correspondence  $\varphi$  from the isolated segments, the connected components of the detached surface, and the boundary components of each connected component of the detached surface of  $\mathcal{C}$  to those of  $\mathcal{C}'$  that preserves the genus, the orientability, the incidences, and the cyclic ordering of the singular points on each boundary component. More precisely, for the latter point: for each connected component  $C$  of the detached surface of  $\mathcal{C}$ , if  $C$  is orientable, then the lists of singular points appearing on each boundary component

of  $C$  and  $\varphi(C)$  are identical up to global reversal of all these cyclic orderings simultaneously, corresponding to a change of the orientation of the connected component; if  $C$  is non-orientable, then the lists of singular points appearing on each boundary component of  $C$  and  $\varphi(C)$  are identical up to the possible individual reversal of some of these cyclic orderings. The proof is tedious but straightforward, and the existence of an isomorphism can obviously be tested in the indicated time.  $\square$

#### 7.2.4 Proper and cellular graph embeddings on 2-complexes

Let  $\mathcal{C}$  be a 2-complex with size  $c$ ,  $G$  a graph, and  $\Gamma$  an embedding of  $G$  on  $\mathcal{C}$ . The embedding  $\Gamma$  is *proper* if:

- the image of  $\Gamma$  meets the boundary of  $\mathcal{C}$  only on singular points;
- the vertices of  $\Gamma$  cover the singular points of  $\mathcal{C}$ .

The embedding  $\Gamma$  is *cellular* if each face of  $\Gamma$  is an open disk plus possibly some part of the boundary of  $\mathcal{C}$ . We emphasize that this definition slightly departs from the standard one. Moreover, we will only consider cellular embeddings that are proper.

Traditional data structures for graphs on surfaces handle graphs embedded cellularly; *rotation systems* [MT01] constitute one example of such a data structure. In order to have efficient algorithms, refined data structures, e.g., with the gem representation [Epp03, Section 2], are needed. The basic element in the gem representation is the *flag*, an incidence between a vertex, an edge, and a face of the graph. Three involutions allow to move from each flag to a nearby flag. Each flag contains a pointer to the underlying vertex, edge, and face.

One can easily extend such data structures to possibly non-cellular embeddings on surfaces [CM14, Section 2.2]. In this framework, one must store the topology of each face, which is not necessarily homeomorphic to a disk. Also, a face may have several boundary components; two-way pointers connect each face to one flag of each boundary component (or to an isolated vertex of the graph, if that boundary component is a single vertex); if a face is orientable and has several boundary components, then these pointers must induce a consistent orientation of these boundary cycles. It is important to remark that this data structure also allows to recover the topology of the underlying surface.

Let  $\Gamma$  be a proper graph embedding of a graph  $G$  on a 2-complex  $\mathcal{C}$  (under the assumptions of Proposition 7.2.1). Let  $\mathcal{S}$  be the detached surface of  $\mathcal{C}$ . Because  $\Gamma$  is proper, it naturally induces an embedding  $\Gamma'$ , of another graph  $G'$ , on  $\mathcal{S}$ ; some vertices of  $G$  located on singular points of  $\mathcal{C}$  are duplicated in  $G'$ , the vertices of  $G$  located in the relative interior of isolated segments are absent from  $G'$ , and the edges of  $G$  not in  $G'$  are edges on the isolated segments of  $\mathcal{C}$ . Our data structure, called *combinatorial map*, for storing the graph embedding  $\Gamma$  and the 2-complex  $\mathcal{C}$  consists of storing (1) the graph embedding  $\Gamma'$  on  $\mathcal{S}$ , as indicated in the previous paragraph, (2) the isolated segments of  $\mathcal{C}$ , together with, for each such isolated segment, an ordered list alternating vertices and edges of  $\Gamma$  (or, instead of an edge, a mark indicating the absence of such an edge in

the region of the isolated segment between the incident vertices), (3) the identifications of vertices of  $\Gamma'$  that are needed to recover  $\Gamma$  (and thus implicitly  $\mathcal{C}$ ).

Isomorphisms between combinatorial maps are defined in the obvious way, similar to the concept of isomorphism between topological data structures: Two combinatorial maps are isomorphic if there is an isomorphism between the combinatorial maps restricted to the detached surfaces, isomorphisms between the maps on each isolated segments, and such that incidences are preserved on the singular points. We can easily test isomorphism between two combinatorial maps of size  $k$  and  $k'$ , respectively, in  $(k + k')^{O(k+k')}$  time.

We will need an algorithm to enumerate all proper embeddings of small graphs on a given 2-complex. This is achieved in the following lemma.

**Lemma 7.2.3.** *Let  $\mathcal{C}$  be a 2-complex of size  $c$  and  $k$  an integer. We can enumerate the  $(c + k)^{O(c+k)}$  combinatorial maps of graphs with at most  $k$  vertices and at most  $k$  edges properly embedded on  $\mathcal{C}$  in  $(c + k)^{O(c+k)}$  time.*

*Proof.* The strategy is the following. In the first step, we enumerate a set of proper graph embeddings on some 2-complexes, which necessarily contains all the desired combinatorial maps. In the second step, we prune this set to keep only the desired combinatorial maps, by eliminating those that contains too many vertices or edges, or that correspond to an embedding on a 2-complex not homeomorphic to  $\mathcal{C}$ .

*First step.* Let  $\Gamma$  be a proper embedding of a graph with at most  $k$  vertices and  $k$  edges on  $\mathcal{C}$ . Let  $\mathcal{S}$  be the detached surface associated to  $\mathcal{C}$ ; this surface is possibly disconnected and has genus at most  $c$ . The image of  $\Gamma$  on  $\mathcal{S}$  is a graph with at most  $k + c$  vertices and  $k$  edges.

We first enumerate, in a possibly redundant way, the set  $M_1$  of combinatorial maps of cellular graph embeddings with at most  $k + c$  vertices and  $k$  edges on a possibly disconnected surface without boundary. There are  $2^{O(c+k)}$  such combinatorial maps, which can be enumerated in  $2^{O(c+k)}$  time, for example using rotation systems. Some vertices may be isolated, if the corresponding connected component of  $\mathcal{S}$  is a sphere.

By simplifying the surface, every graph embedding can be transformed into a cellular graph embedding: remove each face and paste a disk to each cycle of the graph that was a boundary component of a face. Conversely, every (possibly non-cellular) graph is obtained from some cellular one by (1) connecting some faces together (creating a face of genus zero with several boundary components) and (2) adding some genus and non-orientability to some faces. So, for each map in  $M_1$ , we perform all these operations in all possible ways, by putting genus at most  $c$  in each face. For each such map in  $M_1$ , there are  $(c + k)^{O(c+k)}$  possibilities. We thus obtain, in  $(c + c)^{O(c+k)}$  time, a set  $M_2$  of  $(c + k)^{O(c+k)}$  combinatorial maps on surfaces, and the set  $M_2$  contains all combinatorial maps of (possibly non-cellular) graph embeddings on surfaces of genus at most  $c$ .

Finally, we add at most  $c$  isolated segments, choose how endpoints of these isolated segments and vertices of the embedding on the detached surface are identified, and decide how each isolated segment is covered by the embedding. There are  $(c+k)^{O(c+k)}$  ways to do this. We thus have computed, in  $(c+k)^{O(c+k)}$  time, a set  $M$  of  $(c+k)^{O(c+k)}$  combinatorial



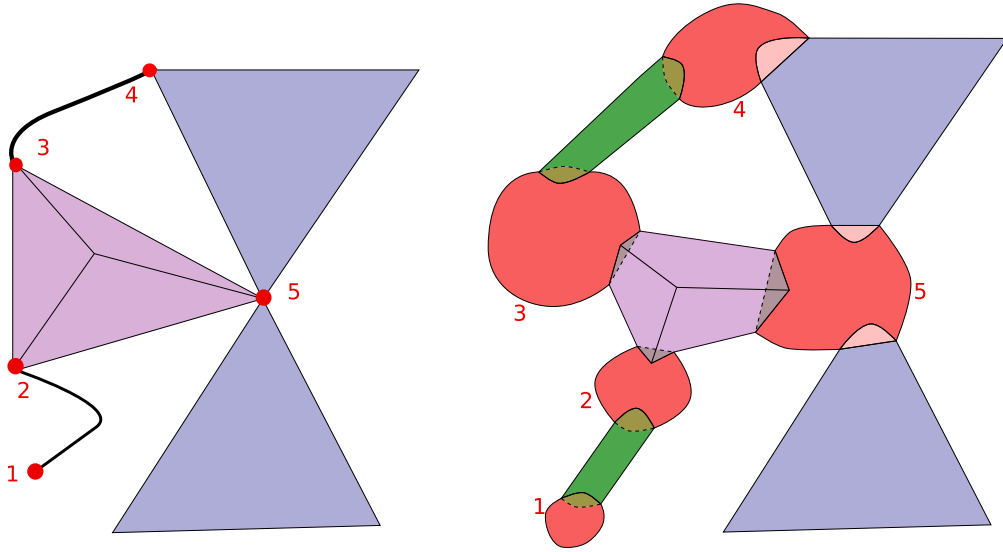


Figure 7.2: On the left: The same 2-complex as Figure 7.1. On the right: a supersurface of  $\mathcal{C}$  as used in Lemma 7.2.4.

maps of graphs on 2-complexes, which contains all the combinatorial maps indicated in the statement of the lemma.

*Second step.* First, we easily discard the combinatorial maps in  $M$  containing more than  $k$  vertices or  $k$  edges. Then, we discard the maps in  $M$  corresponding to a 2-complex different from  $\mathcal{C}$ . For this purpose, for each map  $m$  in  $M$ , we iteratively remove the edges of the graph embedding, preserving the underlying 2-complex. When removing an edge from the detached surface, the topology of the incident face(s) change; we preserve this information. Finally, we remove every isolated vertex that does not lie on a singular point of the 2-complex. The data structure that we have now is essentially the one that is described in Section 7.2.3; we can thus easily decide whether that 2-complex is homeomorphic to  $\mathcal{C}$  (Lemma 7.2.2), and discard  $m$  if and only if it is not the case.

Finally, and although this is not strictly needed, we can easily remove the duplicates in the combinatorial maps, by testing pairwise isomorphism between these maps.  $\square$

### 7.2.5 Graphs embeddable on a fixed 2-complex have bounded genus

**Lemma 7.2.4.** *Let  $\mathcal{C}$  be a 2-complex without 3-book. Let  $c$  be either the size of  $\mathcal{C}$  or its number of simplices. Every graph embeddable on  $\mathcal{C}$  is embeddable on a surface of genus at most  $10c$ .*

*Proof.* Let  $\mathcal{S}$  be a supersurface of  $\mathcal{C}$ . Recall that any graph embeddable on  $\mathcal{C}$  is embeddable on  $\mathcal{S}$ . It thus suffices to prove that the genus of  $\mathcal{S}$  is at most  $10c$ . Indeed, if  $c$  is the size of  $\mathcal{C}$ , this follows from Euler's formula (intuitively, the number of "handles" created is at most the number of link components). If  $c$  is the number of simplices of  $\mathcal{C}$ , it follows from the fact that the total genus of the detached surface is at most  $c$ , again

by Euler's formula, and from the fact that each isolated edge or triangle contributes to an increase of at most six for the genus in the construction above.  $\square$

### 7.3 Partitioning graphs

Let  $\mathcal{C}$  be a 2-complex and  $G$  a graph, which satisfy the properties of Proposition [7.2.1](#). In this section, we lay the structural foundations of the dynamic programming algorithm, described in the next section (Proposition [7.4.1](#)). The goal, in this section and the following one, is to obtain an algorithm that takes as input  $\mathcal{C}$  and  $G$ , and, in time FPT in the size of  $\mathcal{C}$  and the branchwidth of  $G$ , reports correctly one of the following two statements:

- $G$  has no proper cellular embedding on  $\mathcal{C}$ ,
- $G$  has an embedding on  $\mathcal{C}$ .

This algorithm uses dynamic programming on a rooted branch decomposition of  $G$ . When processing a node of the rooted branch decomposition, it considers embeddings of the subgraph of  $G$  induced by the edges in the leaves of the subtree rooted at that node in a region of  $\mathcal{C}$ . This region will be delimited by a *partitioning graph* embedded on  $\mathcal{C}$ . Our dynamic program will roughly guess the partitioning graph at each node of the rooted branch decomposition. For this purpose, we need that, if  $G$  has a proper cellular embedding on  $\mathcal{C}$ , it has such an embedding that is *sparse*: at each node of the rooted branch decomposition of  $G$ , the partitioning graph corresponding to the embedding of the induced subgraph is small (its size is upper-bounded by a function of the branchwidth of  $G$  and of the size of  $\mathcal{C}$ ). The goal of this section is to prove that this is indeed the case.

Let  $(E_1, \dots, E_k)$  be an (ordered) partition of the edge set  $E(G)$  of  $G$ . (We will only use the cases  $k = 2$  or  $k = 3$ .) The *middle set* of  $(E_1, \dots, E_k)$  is the set of vertices of  $G$  whose incident edges belong to at least two sets  $E_i$ .

Let  $\Gamma$  be a proper cellular embedding of  $G$  on  $\mathcal{C}$ . Since  $\Gamma$  is cellular, every boundary of  $\mathcal{C}$  is incident to at least one vertex of  $\Gamma$ . Let  $\hat{\Gamma}$  be obtained from  $\Gamma$  by adding edges as follows: for any pair of vertices  $u$  and  $v$  of  $\Gamma$  consecutive along a given boundary component of  $\mathcal{C}$ , we connect  $u$  and  $v$  via a new edge that runs along the boundary component. For each (ordered) partition  $(E_1, \dots, E_k)$  of the edge set of  $G$ , we let  $\hat{E}_1$  be the union of  $E_1$  and of these new edges, and  $\hat{E}_i = E_i$  for each  $i \neq 1$ ; thus,  $(\hat{E}_1, \dots, \hat{E}_k)$  is a partition of the set of edges of  $\hat{\Gamma}$ .

The *partitioning graph*  $\Pi(\Gamma, E_1, \dots, E_k)$  (or more concisely  $\Pi$ ) associated to  $\Gamma$  and  $(E_1, \dots, E_k)$  is a graph properly embedded on  $\mathcal{C}$  (but possibly non-cellularly), with labels on its faces, defined as follows:

- The vertex set of  $\Pi$  is the union of the singular points of  $\mathcal{C}$  and of (the images under  $\Gamma$  of) the middle set of  $E_1, \dots, E_k$ .

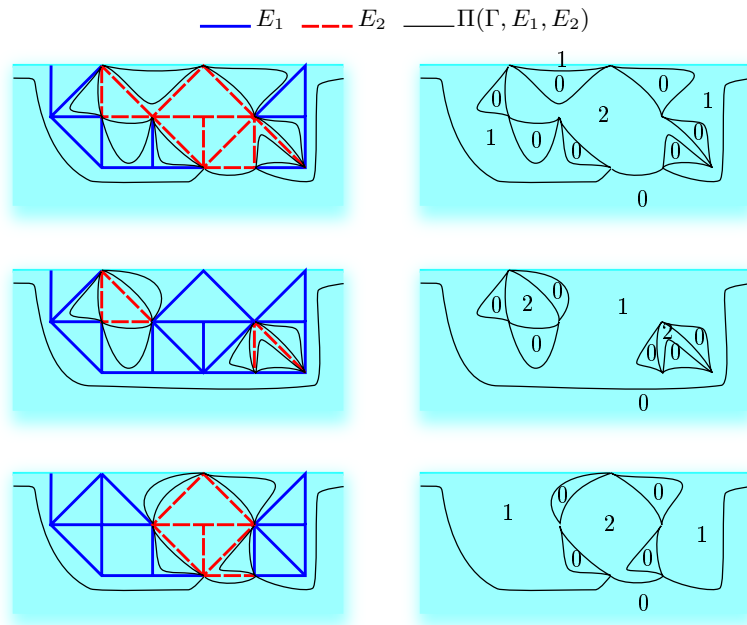


Figure 7.3: Construction of the partitioning graph  $\Pi = \Pi(\Gamma, E_1, E_2)$ , for three choices of the partition  $(E_1, E_2)$  of the same embedding  $\Gamma$ . Only a part of the 2-complex  $\mathcal{C}$  is shown, with a boundary at the upper part, and without singular point. Left: The graph embeddings  $\Gamma$  (in thick lines) and  $\Pi$  (in thin lines). Right: The sole graph  $\Pi$ , together with the labelling of its faces.

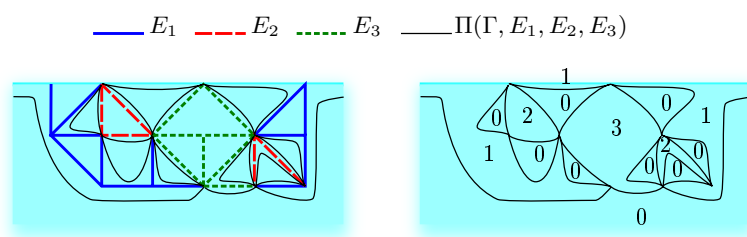


Figure 7.4: The partitioning graph  $\Pi = \Pi(\Gamma, E_1, E_2, E_3)$ . Left: The graph embeddings  $\Gamma$  (in thick lines) and  $\Pi$  (in thin lines). Right: The sole graph  $\Pi$ , together with the labelling of its faces.

- The relative interiors of the edges of  $\Pi$  are disjoint from the edges of  $\hat{\Gamma}$  and from the isolated segments of  $\mathcal{C}$ . Let  $f$  be a face of  $\hat{\Gamma}$  (which is homeomorphic to an open disk plus possibly some points of the boundary of  $\mathcal{C}$ ). Let us describe the edges of  $\Pi$  inside  $f$ .

If, for some  $i \in \{1, \dots, k\}$ , the boundary of  $f$  is comprised only of edges of  $\hat{\Gamma}$  that lie in a single set  $\hat{E}_i$ , then  $\Pi$  contains no edge inside  $f$ . Otherwise, the boundary of  $f$  is a succession of edges of  $\hat{E}_1, \hat{E}_2, \dots, \hat{E}_k$ . The edges of  $\Pi$  inside  $f$  run along the boundary of  $f$ ; for each  $i \in \{1, \dots, k\}$ , for each (maximal) group of consecutive edges in  $\hat{E}_i$  along the boundary of  $f$ , we create an edge of  $\Pi$  that runs along this group, with endpoints the corresponding vertices on the boundary of  $f$  (see Figures 7.3 and 7.4). These vertices either are in the middle set of  $(E_1, \dots, E_k)$ , or lie on the boundary of  $\mathcal{C}$  (and thus on singular points of  $\mathcal{C}$ ).

It follows from the construction that  $\hat{\Gamma}$  and  $\Pi$  intersect only at common vertices.

- Each face of  $\Pi$  is labelled by an integer in  $\{0, \dots, k\}$  as follows: faces of  $\Pi$  containing edges in  $\hat{E}_i$  are labelled  $i$ , and the other ones are labelled 0. By construction of the graph  $\Pi$ , each face of  $\Pi$  contains edges from at most one set  $\hat{E}_i$ , so this labelling is well defined.

Henceforth, we fix a rooted branch decomposition  $\mathcal{B}$  of  $G$ , the root of which is denoted by  $\rho$ . Every arc  $\alpha$  of  $\mathcal{B}$  naturally partitions  $E(G)$  into two parts  $E_1$  and  $E_2$ , in which  $E_1$  is the part on the same side as  $\rho$ ; this (ordered) partition is the *edge partition associated to  $\alpha$* . Recall that  $\Gamma$  is a proper and cellular embedding of  $G$  on  $\mathcal{C}$ ; we let  $\Pi(\Gamma, \alpha)$  be  $\Pi(\Gamma, E_1, E_2)$ . Similarly, every node  $\nu$  of  $\mathcal{B}$  naturally partitions  $E(G)$  into three parts  $E_1, E_2$ , and  $E_3$ , in which  $E_1$  is the part on the same side as  $\rho$ ; this partition is the *edge partition associated to  $\nu$* ; we let  $\Pi(\Gamma, \nu)$  be  $\Pi(\Gamma, E_1, E_2, E_3)$ .

We say that  $\Gamma$  is *sparse* (with respect to  $\mathcal{B}$ ) if the following conditions hold, letting  $c$  be the size of  $\mathcal{C}$  and  $w$  the width of  $\mathcal{B}$ :

- For each arc  $\alpha$  of  $\mathcal{B}$ , the graph  $\Pi(\Gamma, \alpha)$  has at most  $74c + 26w$  edges;
- similarly, for each internal node  $\nu$  of  $\mathcal{B}$ , the graph  $\Pi(\Gamma, \nu)$  has at most  $3(74c + 26w)$  edges.

The result of this section is the following.

**Proposition 7.3.1.** *Let  $\mathcal{C}$  be a 2-complex and  $G$  a graph, which satisfy the properties of Proposition 7.2.1. Let  $\mathcal{B}$  be a rooted branch decomposition of  $G$ . Assume that  $G$  has a proper cellular embedding on  $\mathcal{C}$ . Then it has a proper cellular embedding  $\Gamma$  on  $\mathcal{C}$  that is sparse (with respect to  $\mathcal{B}$ ).*

### 7.3.1 Monogons and bigons

A *monogon* of a graph  $\Pi$  embedded on a 2-complex  $\mathcal{C}$  is a face of  $\Pi$  that is an open disk whose boundary is a single edge of  $\Pi$  (a loop). Similarly, a *bigon* of  $\Pi$  is a face of  $\Pi$

that is an open disk whose boundary is the concatenation of two edges of  $\Pi$  (possibly the same edge appearing twice). The following general lemma on graphs embedded on surfaces without monogons or bigons will be used; some particular cases have been used before [Cha+08, Lemma 2.1].

**Lemma 7.3.2.** *Let  $\mathcal{S}$  be a surface of genus  $g$  without boundary. Let  $\Pi$  be a graph embedded on  $\mathcal{S}$ , not necessarily cellularly. Assume that  $\Pi$  has no monogon or bigon. Then  $|E(\Pi)| \leq \max\{0, 3g + 3|V(\Pi)| - 6\}$ .*

*Proof.* We begin by adding edges to  $\Pi$  as long as it is possible to do so, without introducing any new vertex, monogon, or bigon. Let  $\Pi'$  be the resulting embedded graph. We claim that every face of  $\Pi'$  is a triangle (a disk incident with three edges), except in the following cases:

1.  $\Pi'$  is the empty graph;
2.  $\mathcal{S}$  is a sphere, and  $\Pi'$  has two vertices and no edge;
3.  $\mathcal{S}$  is a sphere, and  $\Pi'$  has a single vertex and no edge;
4.  $\mathcal{S}$  is a projective plane, and  $\Pi'$  has a single vertex and no edge.

Indeed, let  $f$  be a face of  $\Pi'$ . If  $f$  has no boundary component, then since  $\mathcal{S}$  is connected, we are in Case 1 (an isolated vertex would account for a boundary component). Assume that  $f$  has at least two boundary components. We add an edge in  $f$  connecting vertices on these two boundary components. This cannot create any monogon or bigon, except if the two boundary components are both reduced to a single vertex and  $\mathcal{S}$  is a sphere (Case 2). So we can assume that  $f$  has a single boundary component. If  $f$  is orientable and has genus zero, then either we are in Case 3, or  $\mathcal{S}$  is a disk of degree at least four, in which case we can add an edge to split it into smaller disks without creating any monogon or bigon. If  $f$  is orientable and has Euler genus at least two (i.e., orientable genus at least one), we can add an edge that forms a non-separating arc (relatively to the boundary) in  $f$ ; it does not form any monogon or bigon. If  $f$  is non-orientable and has (non-orientable) genus at least two, we can add an edge that forms a separating arc in  $f$ , cutting that surface into two non-orientable surfaces of genus at least one; it does not form any monogon or bigon. Finally, if  $f$  is non-orientable and has (non-orientable) genus one, either the boundary component is reduced to a single vertex, so we are in Case 4, or this face has degree at least one; then it is a Möbius band with at least one vertex and one edge on its boundary, and we can add a loop that is a non-contractible arc (relatively to the boundary) in  $f$ , without forming any monogon or bigon. The only remaining possibility is that  $f$  is a triangle.

It is clear that the statement of the lemma holds whenever we are in one of the four above cases. So we can assume without loss of generality that each face of  $\Pi'$  is a triangle. Since  $V(\Pi) = V(\Pi')$  and  $|E(\Pi)| \leq |E(\Pi')|$ , it suffices to prove the result for  $\Pi'$  instead of  $\Pi$ . Double-counting the incidences between edges and faces implies that the number of triangles  $\tau$  satisfies  $3\tau = 2|E(\Pi')|$ ; plugging this into Euler's formula implies that  $|V(\Pi')| - |E(\Pi')|/3 = 2 - g$ , so  $|E(\Pi')| = 3g + 3|V(\Pi')| - 6$ , as desired.  $\square$

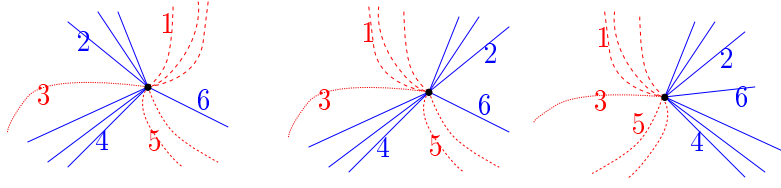


Figure 7.5: Left: A vertex with 6 intervals, numbered from 1 to 6. Middle: The cyclic order obtained by applying the first type of simplification operation on intervals 1 and 2. After the simplification, the intervals 1 and 3 are merged into a single one, and similarly for the intervals 2 and 6. Right: The cyclic order obtained by applying the second type of simplification to the configuration on the left, on pairs of intervals  $\{1, 2\}$  and  $\{4, 5\}$ . After the simplification, the intervals 1, 3, and 5 are merged, and similarly for the intervals 2, 6, and 4.

### 7.3.2 Vertex simplifications

The proof of Proposition [7.3.1](#) starts with any proper cellular embedding of  $\Gamma$ , and iteratively changes the cyclic ordering of edges around vertices in a specific way. Let  $(E_1, E_2)$  be an (ordered) partition of  $E(G)$ , let  $v$  be a vertex of  $G$ , and let  $C$  be a link component at  $v$  (if the image of  $v$  under  $\Gamma$  is a singular point, there may be several such link components). We restrict our attention to the edges of  $\hat{\Gamma}$  incident to  $v$  and belonging to  $C$ , in cyclic order around  $v$ . For  $i = 1, 2$ , an *interval* (at  $v$ , relatively to  $(\hat{E}_1, \hat{E}_2)$ ) is a maximal contiguous subsequence of edges in this cyclic ordering that all belong to  $\hat{E}_i$ ; the interval is labelled  $i$ . *Simplifying*  $v$  (with respect to  $(E_1, E_2)$ ) means changing the cyclic ordering of the edges of  $\hat{\Gamma}$  incident to  $v$  in  $C$  by one of the two following operations (Figure [7.5](#)):

1. either exchanging two consecutive intervals in that ordering, in a way that the ordering of the edges in each interval is preserved; this operation is allowed only if  $v$  is incident to at least four intervals;
2. or performing the previous operation twice, on two disjoint pairs of consecutive intervals in that ordering; this is allowed only if  $v$  is incident to at least six intervals.

We will rely on the following lemma.

**Lemma 7.3.3.** *Let  $\Gamma$  be a proper cellular embedding of  $G$  on  $\mathcal{C}$ , and let  $(E_1, E_2)$  be an (ordered) partition of  $E(G)$ . Let  $\Gamma'$  be another proper cellular embedding of  $G$ , obtained from  $\Gamma$  by simplifying one or two vertices with respect to  $(E_1, E_2)$ , while keeping the other cyclic orderings unchanged. Then:*

1.  $|E(\Pi(\Gamma', E_1, E_2))| < |E(\Pi(\Gamma, E_1, E_2))|$ ;
2. for each (ordered) partition  $(\tilde{E}_1, \tilde{E}_2)$  of  $E(G)$  such that  $\hat{E}_i \subseteq \hat{E}_j$  for some  $i, j \in \{1, 2\}$ , we have  $|E(\Pi(\Gamma', \tilde{E}_1, \tilde{E}_2))| \leq |E(\Pi(\Gamma, \tilde{E}_1, \tilde{E}_2))|$ .

*Proof.* The proof is based on the following easy but key observations (the second one will be reused later):

- A simplification of  $v$  strictly decreases the number of intervals at  $v$ ;
- the number of half-edges of  $\Pi(\Gamma, E_1, E_2)$  at  $v$  in the link component  $C$  equals twice the number of intervals associated to  $(\hat{E}_1, \hat{E}_2)$  at  $v$  in  $C$ .

The first point of the lemma immediately follows. For the second point, let us consider, in the cyclic ordering around  $v$  in  $C$ , a maximal contiguous sequence of edges in  $\hat{E}_i$ . Since  $\hat{E}_i \subseteq \hat{E}_j$ , when simplifying with respect to  $(E_1, E_2)$ , this sequence is still contiguous in the new embedding  $\Gamma'$ . It follows that the number of intervals associated to  $(\hat{E}_1, \hat{E}_2)$  does not increase when replacing  $\Gamma$  with  $\Gamma'$ .  $\square$

### 7.3.3 Rearranging $\Gamma$ with respect to an edge partition

We can now prove the following lemma:

**Lemma 7.3.4.** *Let  $\Gamma$  be a proper cellular embedding of  $G$  on  $\mathcal{C}$ , and let  $(E_1, E_2)$  be an (ordered) partition of  $E(G)$ . There exists a proper cellular embedding  $\Gamma'$  of  $G$  such that:*

- $|E(\Pi(\Gamma', E_1, E_2))| \leq 74c + 26w$ , where  $w$  is the size of the middle set of  $(E_1, E_2)$ ;
- for each (ordered) partition  $(\tilde{E}_1, \tilde{E}_2)$  of  $E(G)$  such that  $\hat{E}_i \subseteq \hat{E}_j$  for some  $i, j \in \{1, 2\}$ , we have  $|E(\Pi(\Gamma', \tilde{E}_1, \tilde{E}_2))| \leq |E(\Pi(\Gamma, \tilde{E}_1, \tilde{E}_2))|$ .

*Proof.* Here is an overview of the proof. Let  $\Pi := \Pi(\Gamma, E_1, E_2)$ . We will assume that  $\Pi$  has “many monogons or bigons” (in a sense made precise below) and show that there is another cellular embedding  $\Gamma'$  of  $G$  such that:

- $|E(\Pi(\Gamma', E_1, E_2))| < |E(\Pi(\Gamma, E_1, E_2))|$ ;
- for each (ordered) partition  $(\tilde{E}_1, \tilde{E}_2)$  of  $E(G)$  such that  $\hat{E}_i \subseteq \hat{E}_j$  for some  $i, j \in \{1, 2\}$ , we have  $|E(\Pi(\Gamma', \tilde{E}_1, \tilde{E}_2))| \leq |E(\Pi(\Gamma, \tilde{E}_1, \tilde{E}_2))|$ .

By repeatedly iterating this argument, and up to replacing  $\Gamma$  with  $\Gamma'$ , this implies that we can assume without loss of generality that  $\Pi$  has “not too many monogons or bigons”. We will then show that this latter property implies that  $\Pi$  has at most  $74c + 26w$  edges, which concludes.

First, let  $v$  be a vertex of  $\Pi$ , and let  $C$  be a link component of  $\mathcal{C}$  at  $v$  in  $\Pi$ . Assume that  $v$  has at least 8 incident half-edges in  $C$  (and thus at least four intervals), and that  $\Pi$  has a monogon incident to  $v$  in  $C$ . By construction of  $\Pi$ , the monogon is not labelled 0. Thus, a non-empty subgraph of  $\Gamma$  lies inside the monogon, attached to the rest of  $\Gamma$  only by  $v$ , and corresponds to an interval  $s$  of  $\Gamma$  at  $v$  in  $C$ . In  $\Gamma$ , we move the part of  $\Gamma$  that lies inside the monogon on the other side of the edges comprising an adjacent interval  $s'$ ; see Figure [7.6](#). This simplifies  $v$  by swapping  $s$  with  $s'$ , because  $\Gamma$  has at least 4 intervals at  $v$

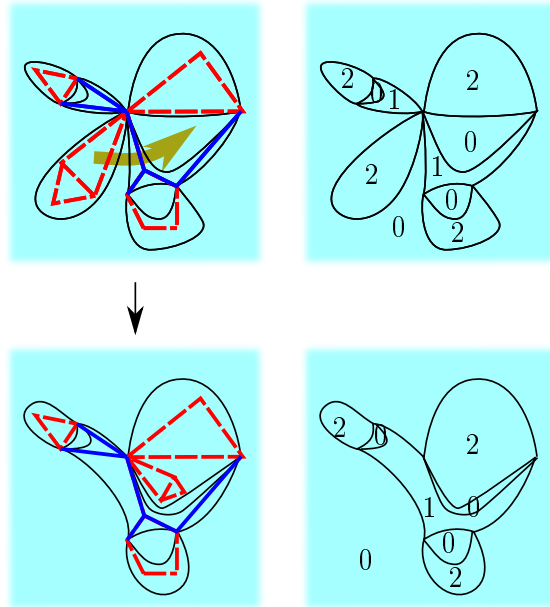


Figure 7.6: Decreasing the number of monogons in  $\Pi$ .

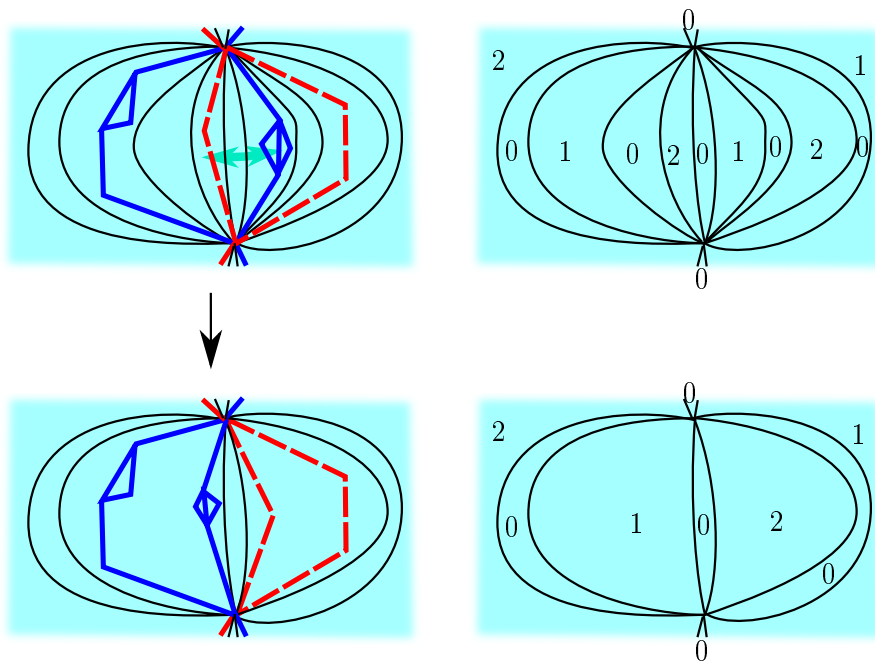


Figure 7.7: Decreasing the number of bigons in  $\Pi$ .



in  $C$ . Note that there is no singular point in the interior of the monogon, because there would be a vertex of  $\Pi$  located on the singular point. The resulting graph embedding  $\Gamma'$  is still proper and cellular, and satisfies the desired properties, by Lemma [7.3.3](#).

Now, let us assume that  $\Pi$  contains a sequence of bigons  $B_1, \dots, B_8$  such that  $B_i$  and  $B_{i+1}$  share an edge for each  $i$ . So without loss of generality, we can assume that  $B_1$  and  $B_5$  are labelled 1,  $B_3$  and  $B_7$  are labelled 2, and the other bigons are labelled 0. We modify  $\Gamma$  by exchanging the parts of  $\Gamma$  inside  $B_3$  and  $B_5$ ; see Figure [7.7](#). The resulting embedding  $\Gamma'$  is also proper and cellular. This operation simplifies the endpoints  $u$  and  $v$  of these bigons. (If  $u = v$ , this is Case 2 from the definition of a simplification.)

We can iterate the above procedures, but only finitely many times because  $|E(\Pi(\Gamma, E_1, E_2))|$  strictly decreases at each step. We have proved that, up to changing our initial embedding  $\Gamma$ , we can assume without loss of generality the following: (i) Let  $v$  be a vertex of  $\Pi$  that is incident to a monogon of  $\Pi$  in a link component  $C$ ; then  $v$  is incident to at most 7 half-edges in  $C$ , and thus at most 4, since this number is a multiple of 4 by construction; (ii)  $\Pi$  has no sequence of 8 consecutive bigons as above. To conclude, it suffices to prove that any graph  $\Pi$  satisfying these conditions has at most  $74c + 26w$  edges.

We modify  $\Pi$  by removing all monogons, and then by iteratively replacing each bigon with a single edge, when the edges bounding the bigon are distinct. The removal of monogons does not create any sequence of 8 consecutive bigons, because monogons are attached to vertices of degree at most 4 in their link component. So in the first step, for each vertex  $v$  and each link component  $C$  of  $v$ , at most two monogons in  $C$  attached to  $v$  are removed; the number of such monogons is at most  $2w$  (for the vertices of  $\Pi$  not on a singular point of  $\mathcal{C}$ ) plus  $2c$  (for the vertices of  $\Pi$  on singular points of  $\mathcal{C}$ ). In the second step, the number of edges is divided by at most 8. Thus, if  $\Pi'$  denotes the new embedding, we have:

$$|E(\Pi)| \leq 2(c + w) + 8|E(\Pi')|. \quad (7.1)$$

We now bound the number of edges of  $\Pi'$ . For this purpose, let  $\mathcal{S}$  be the detached surface of  $\mathcal{C}$ , and let  $\Pi''$  be the graph naturally corresponding to  $\Pi'$  on  $\mathcal{S}$  (see Section [7.2.4](#)). Any bigon of  $\Pi''$  whose boundary consists of the same edge repeated twice is itself a connected component of  $\mathcal{S}$ : either a sphere, in which case the corresponding connected component of  $\Pi''$  is made of two vertices and a single edge, or a projective plane, in which case the corresponding connected component of  $\Pi''$  is made of a single vertex and a single edge. Thus, in these connected components, the number of edges of  $\Pi''$  is at most the number of vertices. Let  $\mathcal{S}_0$  be obtained from  $\mathcal{S}$  by removing these connected components, and  $\Pi''_0$  the restriction of  $\Pi''$  to  $\mathcal{S}_0$ .

$\Pi''_0$  has no monogon or bigon. Let  $\tilde{\mathcal{S}}_0$  be obtained from  $\mathcal{S}_0$  by attaching a handle to each boundary component; it has a natural cellular graph embedding with at most  $2c$  edges, and thus genus at most  $2c$ ; the graph  $\Pi''_0$  corresponds to an embedding of a graph  $\bar{\Pi}''_0$  on  $\tilde{\mathcal{S}}_0$ , with no monogon or bigon. Lemma [7.3.2](#) applied to the restriction of  $\bar{\Pi}''_0$  to each connected component of  $\tilde{\mathcal{S}}_0$  implies that  $|E(\Pi''_0)| \leq 6c + 3|V(\bar{\Pi}''_0)|$ .

Thus,  $|E(\Pi')| = |E(\Pi''_0)| \leq 6c + 3|V(\Pi''_0)|$ . Moreover,  $|V(\Pi''_0)| \leq c + w$ . Now, Inequality [\(7.1\)](#) implies that  $|E(\Pi)| \leq 2(c + w) + 8(6c + 3(c + w)) = 74c + 26w$ , as desired.  $\square$

### 7.3.4 Proof of Proposition 7.3.1

*Proof of Proposition 7.3.1.* Let  $\mathcal{B}$  be a rooted branch decomposition of  $G$ , and let  $\Gamma$  be a proper cellular embedding of  $G$  on  $\mathcal{C}$ . We consider each arc  $\alpha$  of the rooted branch decomposition in turn, in an arbitrary order. For each such arc, we modify  $\Gamma$  by applying Lemma 7.3.4. We first claim that after these iterations, for each arc  $\alpha$  of  $\mathcal{B}$ , we have  $|E(\Pi(\Gamma, \alpha))| \leq 74c + 26w$ .

First, immediately after applying the above procedure to an arc  $\tilde{\alpha}$  of  $\mathcal{B}$ , corresponding to the (ordered) partition  $(\tilde{E}_1, \tilde{E}_2)$  of  $E(G)$ , we have  $|E(\Pi(\Gamma, \tilde{E}_1, \tilde{E}_2))| \leq 74c + 26w$ . We now prove that subsequent applications of Lemma 7.3.4 to other arcs of the rooted branch decomposition do not increase this number of edges. Indeed, let  $\alpha$  be another arc, corresponding to the (ordered) partition  $(E_1, E_2)$  of  $E(G)$ , to which we apply Lemma 7.3.4. The arc  $\alpha$  partitions the nodes of the tree  $\mathcal{B}$  into two sets  $N_1$  and  $N_2$ , and similarly  $\tilde{\alpha}$  partitions the nodes of the tree  $\mathcal{B}$  into two sets  $\tilde{N}_1$  and  $\tilde{N}_2$ . Because  $\mathcal{B}$  is a tree, we have  $\tilde{N}_i \subseteq N_j$  for some  $i, j \in \{1, 2\}$ . This implies that  $\tilde{E}_i \subseteq \hat{E}_j$  for some  $i, j \in \{1, 2\}$ ; so the second item of Lemma 7.3.4 implies that the number of edges of  $\Pi(\Gamma, \tilde{E}_1, \tilde{E}_2)$  does not increase when processing arc  $\alpha$ . This proves the claim.

Finally, there remains to prove that, for each internal node  $\nu$  of  $\mathcal{B}$ , the graph  $\Pi(\Gamma, \nu)$  has at most  $3(74c + 26w)$  edges. Let  $(E_1, E_2, E_3)$  be the edge partition associated with  $\nu$ . By the claim we have just proved, it suffices to prove that the number of edges of this graph is at most the sums of the numbers of edges of  $\Pi(\Gamma, E_1, E_2 \cup E_3)$ ,  $\Pi(\Gamma, E_1 \cup E_2, E_3)$ , and  $\Pi(\Gamma, E_1 \cup E_3, E_2)$ .

Let  $v$  and  $C$  be as above. We look at the cyclic ordering, around  $v$  at  $C$ , of the half-edges of  $\hat{\Gamma}$  (ignoring the possible boundary component of  $\mathcal{C}$  that may arise once in this cyclic ordering). Between two such consecutive half-edges, there are either zero or two half-edges in  $\Pi(\Gamma, \nu)$ . In the latter case, this means that these two consecutive half-edges of  $\Gamma$  are in two different sets  $E_i$  and  $E_j$ . Thus, between these two consecutive half-edges of  $\Gamma$ , necessarily two half-edges appear in at least one of  $\Pi(\Gamma, E_1, E_2 \cup E_3)$ ,  $\Pi(\Gamma, E_1 \cup E_2, E_3)$ , and  $\Pi(\Gamma, E_1 \cup E_3, E_2)$ . This concludes the proof.  $\square$

## 7.4 Dynamic programming algorithm

The result of this section is the following proposition.

**Proposition 7.4.1.** *Let  $\mathcal{C}$  be a 2-complex and  $G$  a graph, which satisfy the properties of Proposition 7.2.1. Let  $c$  be the size of  $\mathcal{C}$  and  $n$  the number of vertices and edges of  $G$ . Let  $\mathcal{B}$  be a rooted branch decomposition of  $G$  of width  $w$ . In  $(c + w)^{O(c+w)}n$  time, one can report one of the following statements, which is true:*

- $G$  has no sparse proper cellular embedding into  $\mathcal{C}$ ;
- $G$  has an embedding into  $\mathcal{C}$ .

(Proposition 7.3.1 implies that we can remove the adjective “sparse” in the above proposition.)

### 7.4.1 Bounding graphs

Let  $\mathcal{B}$  be a rooted branch decomposition of  $G$  of width  $w$ . Recall (see Section 3.5.2) that the root  $\rho$  of  $\mathcal{B}$  is a leaf associated to no edge of  $G$ . Our algorithm will use dynamic programming in the rooted branch decomposition. For each arc  $\alpha$  of  $\mathcal{B}$ , let  $G_\alpha$  be the subgraph of  $G$  induced by the edges of  $G$  corresponding to the leaves of the subtree of  $\mathcal{B}$  rooted at  $\alpha$ . The general idea is that we compute all possible relevant embeddings of  $G_\alpha$  in subregions of  $\mathcal{C}$ . Such subregions will be delimited by a graph embedded on  $\mathcal{C}$  of small complexity. For the dynamic program to work, we also need to keep track of the location of the vertices in the middle set of  $\alpha$ . More precisely, a **bounding graph** for  $G_\alpha$  is a proper labelled graph embedding  $\Pi$  on  $\mathcal{C}$  (but possibly non-cellular), such that:

- some vertices of  $\Pi$  are labelled; these labels are exactly the vertices of the middle set associated with  $\alpha$ , and each label appears exactly once;
- each unlabelled vertex of  $\Pi$  is mapped to a singular point of  $\mathcal{C}$ ;
- each face of  $\Pi$  is labelled 0, 1, or 2;
- $G_\alpha$  has an embedding  $\Gamma_\alpha$  that **respects**  $\Pi$ : each vertex of  $\Pi$  labelled  $v$  is mapped, under  $\Pi$ , to the image of  $v$  in  $\Gamma_\alpha$ ; moreover, the relative interior of each edge of  $\Gamma_\alpha$  lies in the interior of a face of  $\Pi$  labelled 2.

(It may seem strange to require that each singular point of  $\mathcal{C}$  be covered by a vertex of  $\Pi$ ; however, it is necessary to cover at least the singular points on the boundary of  $\mathcal{C}$ , and this more general requirement, albeit slightly artificial, simplifies the argumentation. Also, it is slightly simpler to have three labels for the faces of a bounding graph, although two would suffice.)

A bounding graph for  $G_\alpha$  is **sparse** if it has at most  $74c + 26w$  edges. Remark that, if  $\Gamma$  is a sparse proper cellular embedding of  $G$  on  $\mathcal{C}$  (as defined in Section 7.3), then  $\Pi(\Gamma, \alpha)$  is a sparse bounding graph for the restriction of  $\Gamma$  to  $G_\alpha$ .

Henceforth, we regard two (labelled) properly embedded graphs as equal if and only if their (labelled) combinatorial maps are isomorphic. A list  $\mathcal{L}_\alpha$  of sparse bounding graphs for  $G_\alpha$  is **exhaustive** if the following condition holds: If  $G$  has a sparse proper cellular embedding on  $\mathcal{C}$ , then for each such embedding  $\Gamma$ , the (combinatorial map of the) graph  $\Pi(\Gamma, \alpha)$  is in  $\mathcal{L}_\alpha$ .

The induction step for the dynamic programming algorithm is the following.

**Proposition 7.4.2.** *Let  $\nu$  be a non-root node of  $\mathcal{B}$  and  $\alpha$  be the arc of  $\mathcal{B}$  incident to  $\nu$  that is the closest to the root  $\rho$ . Assume that, for each arc  $\beta \neq \alpha$  of  $\mathcal{B}$  incident to  $\nu$ , we are given an exhaustive list of sparse bounding graphs for  $G_\beta$ . Then we can, in  $(c + w)^{O(c+w)}$  time, compute an exhaustive list of  $(c + w)^{O(c+w)}$  sparse bounding graphs for  $G_\alpha$ .*

Assuming Proposition 7.4.2, the proof of which is deferred to the next subsection, it is easy to prove Proposition 7.4.1:

*Proof of Proposition 7.4.1, assuming Proposition 7.4.2.* We apply the algorithm of Proposition 7.4.2 in a bottom-up manner in the rooted branch decomposition  $\mathcal{B}$ . Let  $\alpha$  be the arc of  $\mathcal{B}$  incident with the root node  $\rho$ . We end up with an exhaustive list of sparse bounding graphs for  $G_\alpha = G$ . By definition of a bounding graph, if this list is non-empty, then  $G$  has an embedding on  $\mathcal{C}$ . On the other hand, by definition of an exhaustive list, if this list is empty, then  $G$  has no sparse proper cellular embedding on  $\mathcal{C}$ .

There are  $O(n)$  recursive calls, each of which takes  $(c+w)^{O(c+w)}$  time.  $\square$

### 7.4.2 The induction step: Proof of Proposition 7.4.2

*Proof of Proposition 7.4.2. First case.* Let us first assume that  $\nu$  is a (non-root) leaf of  $\mathcal{B}$ ; thus,  $G_\alpha$  is a single edge  $uv$ . We will compute *all* the labelled combinatorial maps of sparse bounding graphs for  $G_\alpha$ . It is clear that this will be an exhaustive list. Indeed, assume that  $G$  has a sparse proper cellular embedding  $\Gamma$  on  $\mathcal{C}$ ; by sparsity,  $\Pi(\Gamma, \alpha)$  has at most  $74c + 26w$  edges; thus, one of the labelled combinatorial maps computed will be equal to that of  $\Pi(\Gamma, \alpha)$ .

So let us describe how to enumerate all the labelled combinatorial maps of bounding graphs for  $G_\alpha$ . Using Lemma 7.2.3, we enumerate all possible labelled (combinatorial maps of) proper graph embeddings  $\Pi$  on  $\mathcal{C}$  such that:

- two vertices of  $\Pi$  are labelled  $u$  and  $v$ ; the other vertices are unlabelled; the singular points of  $\mathcal{C}$  are covered by the vertices of  $\Pi$ ; conversely, every vertex of  $\Pi$ , except perhaps  $u$  and/or  $v$ , is mapped to singular points of  $\mathcal{C}$ ;
- $\Pi$  has at most  $74c + 26w$  edges;
- each face of  $\Pi$  is labelled 0, 1, or 2;
- $\Pi$  has a face labelled 2 whose boundary contains both vertices  $u$  and  $v$ .

It is clear that these labelled combinatorial maps represent all the sparse bounding graphs for  $G_\alpha$ .

*Second case.* Let us now assume that  $\nu$  is an internal node of  $\mathcal{B}$ . As above, let  $\alpha$  be the arc of  $\mathcal{B}$  incident to  $\nu$  that is the closest to the root  $\rho$ . Let  $\beta$  and  $\gamma$  be the arcs different from  $\alpha$  incident to  $\nu$ . Let  $\mathcal{L}_\beta$  and  $\mathcal{L}_\gamma$  be exhaustive lists of bounding graphs for  $G_\beta$  and  $G_\gamma$ , respectively. Intuitively, every pair of bounding graphs in  $\mathcal{L}_\beta$  and  $\mathcal{L}_\gamma$  that are compatible, in the sense that the regions labelled 2 in each of these two graphs are disjoint, will lead to a bounding graph in  $\mathcal{L}_\alpha$ . This is the motivating idea to our approach. More precisely, we will enumerate labelled combinatorial maps  $\Pi$ , each of which can be “restricted” to two compatible graphs, which are possible bounding graphs for  $G_\beta$  and  $G_\gamma$ . If these two restrictions lie in  $\mathcal{L}_\alpha$  and  $\mathcal{L}_\beta$ , this leads to a graph that is added to  $\mathcal{L}_\alpha$ .

We first introduce some terminology. Let  $\Pi$  be a graph properly embedded on  $\mathcal{C}$  (possibly non-cellularly), with faces labelled 0, 1, 2, or 3, and with labels on some vertices. Let  $i, j, k$  be integers such that  $\{i, j, k\} = \{1, 2, 3\}$ . We will define a graph embedding

$\Pi_{i,j}$  obtained from  $\Pi$  by somehow “merging” faces  $i$  and  $j$ . First, for an illustration, refer back to Figures [7.3](#) and [7.4](#): If  $\Pi$  is the graph embedding depicted on the right of Figure [7.4](#), then the configurations shown on the right of Figure [7.3](#) correspond, from top to bottom, to  $\Pi_{2,3}$ ,  $\Pi_{1,3}$ , and  $(\Pi_{1,2})^-$  (the latter being the graph  $\Pi_{1,2}$  in which each face label 3 is replaced by a 2).

Formally,  $\Pi_{i,j}$  is defined as follows. First, let us replace all face labels  $j$  by  $i$ . Now, for each face  $f$  of  $\Pi$  that is homeomorphic to a disk and labelled 0, we do the following. The boundary of  $f$  is made of edges of  $\Pi$ ; for the sake of the discussion, let us temporarily label each such edge by the label of the face on the other side of  $f$ . If all edges on the boundary of  $f$  are all labelled  $i$ , then we remove all these edges, and  $f$  becomes part of a larger face labelled  $i$ . Otherwise, for each maximal subsequence  $e_1, \dots, e_\ell$  of edges along the boundary of  $f$  that are all labelled  $i$ , we remove each of  $e_1, \dots, e_\ell$ , and replace them with an edge inside  $f$  from the source of  $e_1$  to the target of  $e_\ell$ . Finally, we remove all isolated vertices that do not coincide with a singular point of  $\mathcal{C}$ , and all vertices in the relative interior of an isolated segment that are incident to two faces with the same label.

For any labelled combinatorial map  $\Pi$ , we denote by  $\Pi^-$  the same map where each label 3 on a face is replaced by a 2. The easy but key properties of this construction are the following:

- (i) Assume that  $\Pi_{1,3}$  is a bounding graph for  $G_\beta$  and  $(\Pi_{1,2})^-$  is a bounding graph for  $G_\gamma$ . Then  $\Pi_{2,3}$  is a bounding graph for  $G_\alpha$ .
- (ii) The node  $\nu$  naturally partitions the edge set of  $G$  into three parts, which we denote by  $E_1$  (on the side of  $\alpha$ ),  $E_2$  (on the side of  $\beta$ ), and  $E_3$  (on the side of  $\gamma$ ). Assume that  $G$  has a sparse proper cellular embedding  $\Gamma$  on  $\mathcal{C}$  and that  $\Pi = \Pi(\Gamma, E_1, E_2, E_3)$ . Then:
  - $\Pi(\Gamma, \alpha) = \Pi(\Gamma, E_1, E_2 \cup E_3) = \Pi_{2,3}$ ;
  - $\Pi(\Gamma, \beta) = \Pi(\Gamma, E_1 \cup E_3, E_2) = \Pi_{1,3}$ ;
  - $\Pi(\Gamma, \gamma) = \Pi(\Gamma, E_1 \cup E_2, E_3) = (\Pi_{1,2})^-$ .

Property (ii) is, again, illustrated by Figures [7.3](#) and [7.4](#): If  $(E_1, E_2, E_3)$  is the edge partition depicted on Figure [7.4](#), then the edge partitions depicted on Figure [7.3](#), left, are, respectively,  $(E_1, E_2 \cup E_3)$ ,  $(E_1 \cup E_3, E_2)$ , and  $(E_1 \cup E_2, E_3)$ . As shown above, the corresponding partitioning graphs are respectively  $\Pi_{2,3}$ ,  $\Pi_{1,3}$ , and  $\Pi_{1,2}^-$ .

If  $\Pi$  is the graph embedding depicted on the right of Figure [7.4](#), then the configurations shown on the right of Figure [7.3](#) correspond, from top to bottom, to  $\Pi_{2,3}$ ,  $\Pi_{1,3}$ , and  $(\Pi_{1,2})^-$  (the latter being the graph  $\Pi_{1,2}$  in which each face label 2 is replaced by a 2).

We compute our exhaustive list  $\mathcal{L}_\alpha$  of sparse bounding graphs for  $G_\alpha$  as follows. Initially, let this list be empty. Using Lemma [7.2.3](#), we enumerate all combinatorial maps  $\Pi$  of graphs with at most  $c+3w$  vertices and  $3(74c+26w)$  edges properly embedded on  $\mathcal{C}$  (possibly non-cellularly), with faces labelled 0, 1, 2, or 3, and such that the labels appearing on the vertices are exactly the vertices of the middle set of  $\alpha, \beta$ , or  $\gamma$  (and each label appears exactly once). This takes  $(c+w)^{O(c+w)}$  time. Whenever  $\Pi_{1,3} \in \mathcal{L}_\beta$  and

$(\Pi_{1,2})^- \in \mathcal{L}_\gamma$ , we add  $\Pi_{2,3}$  to  $\mathcal{L}_\alpha$ . Finally, we eliminate duplicates by testing pairwise isomorphism between the labelled combinatorial maps in  $\mathcal{L}_\alpha$ , and remove the graphs that are not sparse or contain vertices that bear a label not in the middle set of  $\alpha$ .

$\mathcal{L}_\alpha$  contains only sparse bounding graphs for  $G_\alpha$ , by (i) above. Moreover, let  $\Gamma$  be a sparse proper cellular graph embedding of  $G$  on  $\mathcal{C}$ . By sparsity, one of the graphs  $\Pi$  enumerated in the previous paragraph is  $\Pi(\Gamma, \nu)$ . By definition of  $\mathcal{L}_\beta$  and  $\mathcal{L}_\gamma$ , we have that  $\Pi(\Gamma, \beta) \in \mathcal{L}_\beta$  and  $\Pi(\Gamma, \gamma) \in \mathcal{L}_\gamma$ , so by (ii) above,  $\Pi(\Gamma, \alpha) \in \mathcal{L}_\alpha$ , which implies that  $\mathcal{L}_\alpha$  is exhaustive.  $\square$

## 7.5 Reduction to proper cellular embeddings

This section is devoted to proving the following result:

**Proposition 7.5.1.** *Let  $\mathcal{C}$  be a 2-complex with at most  $c$  simplices, and  $G$  a graph with at most  $n$  vertices and edges and branchwidth at most  $w$ . Assume that  $G$  and  $\mathcal{C}$  satisfy the properties of Proposition 7.2.1. In  $c^{O(c)} + O(cn)$  time, one can compute a graph  $G'$ , and  $c^{O(c)}$  2-complexes  $\mathcal{C}_i$ , such that:*

1. each  $\mathcal{C}_i$  and  $G'$  satisfy the properties of Proposition 7.2.1;
2.  $G'$  has at most  $5cn$  vertices and  $5cn$  edges, and branchwidth at most  $w$ ;
3. each  $\mathcal{C}_i$  has size at most  $c$ ;
4. if, for some  $i$ ,  $G'$  embeds into  $\mathcal{C}_i$ , then  $G$  embeds into  $\mathcal{C}$ ;
5. if  $G$  embeds into  $\mathcal{C}$ , then for some  $i$ ,  $G'$  has a proper cellular embedding into  $\mathcal{C}_i$ .

We start with auxiliary results. Let  $\mathcal{S}$  be a surface (possibly disconnected, possibly with boundary). A **cutting operation** on  $\mathcal{S}$  consists of cutting it along a simple closed curve, and attaching a disk to the resulting boundary component(s). A cutting operation is **essential** if the simple closed curve is non-contractible.

The following result is not hard and essentially folklore (a related but slightly weaker result is provided by Matoušek et al. [Mat+16, Lemma 3.1]), but we could not find a precise reference.

**Lemma 7.5.2.** *Let  $\mathcal{S}$  be a (connected) surface with genus  $g$ . The number of possibly disconnected surfaces, up to homeomorphism, that can be obtained from  $\mathcal{S}$  by a cutting operation is at most  $g + 3$ , and we can compute them in linear time. Moreover, this cutting operation leads either to a single surface with genus strictly smaller than  $g$ , or to two surfaces, the sum of the genera of which equals  $g$ , and the size of the surface (sum of the number of connected components, total genus, and number of boundary components) increases by at most one.*

*Proof.* This basically follows from the classification of surfaces together with Euler's formula. A cutting operation of  $\mathcal{S}$  along a closed curve  $\gamma$  falls into exactly one of the following three categories:

1. *Case where  $\gamma$  is separating.* The cutting operation on  $\mathcal{S}$  results in two surfaces  $\mathcal{S}_1$  and  $\mathcal{S}_2$ , in which their respective genera  $g_1$  and  $g_2$  satisfy  $g = g_1 + g_2$ . Moreover, if  $\mathcal{S}$  is non-orientable, then at least one of  $\mathcal{S}_1$  or  $\mathcal{S}_2$  is non-orientable. Finally, all pairs of surfaces  $(\mathcal{S}_1, \mathcal{S}_2)$  satisfying these constraints can be obtained as the result of a cutting operation on  $\mathcal{S}$ .
2. *Case where  $\gamma$  is non-separating but two-sided.* This is only possible if  $g \geq 2$ . The cutting operation on  $\mathcal{S}$  results in a single surface  $\mathcal{S}'$  with genus  $g - 2$ . If  $\mathcal{S}$  is orientable, then so is  $\mathcal{S}'$ ; otherwise,  $\mathcal{S}'$  is either orientable or non-orientable (unless of course  $g = 2$ , in which case it is necessarily orientable, or  $g$  is odd, in which case it is necessarily non-orientable). All surfaces  $\mathcal{S}'$  satisfying these constraints can be obtained.
3. *Case where  $\gamma$  is one-sided.* This is only possible if  $\mathcal{S}$  is non-orientable and  $g \geq 1$ . The cutting operation on  $\mathcal{S}$  results in a single surface  $\mathcal{S}'$  with genus  $g - 1$ , orientable or not (unless of course  $g = 1$ , in which case it is orientable, or  $g$  is even, in which case it is non-orientable). All surfaces  $\mathcal{S}'$  satisfying these constraints can be obtained.  $\square$

**Lemma 7.5.3.** *Let  $\mathcal{S}$  be a surface with  $k$  connected components, total genus  $g$ , and with  $b$  boundary components in total. In  $(k + g + b)^{O(g+b)}$  time, we can enumerate all  $(k + g + b)^{O(g+b)}$  possibly disconnected surfaces with boundary, up to homeomorphism, arising from  $\mathcal{S}$  by one or several successive essential cutting operations. These surfaces have  $O(k + g + b)$  connected components and size  $O(k + g + b)$ .*

*Proof.* It is useful to organize the set of all surfaces (possibly disconnected, possibly with boundary) arising by essential cutting operations in a tree with root  $\mathcal{S}$ , in which the children of a node result from a single essential cutting operation. We prove that (1) the depth of the tree is  $O(g+b)$  and that (2) each node of the tree has  $O((k+1)(g+1)(b+1))$  children, which concludes (because by Lemma 7.5.2, the size of a surface increases by at most one by a cutting operation).

Let  $\mathcal{S}'$  be a (possibly disconnected, possibly with boundary) surface resulting from a sequence of essential cutting operations on  $\mathcal{S}$ . By Lemma 7.5.2, the total genus of  $\mathcal{S}'$  is at most  $g$ . Moreover, since we consider only essential cutting operations, each connected component of  $\mathcal{S}'$  either has positive genus or contains at least one boundary component, unless it was itself a connected component of  $\mathcal{S}$ ; so the number of connected components of  $\mathcal{S}'$  is at most  $k + g + b$ .

Let  $\varphi(\mathcal{S}')$  be equal to twice the total genus of  $\mathcal{S}'$  minus its number of connected components. By Lemma 7.5.2, this potential function strictly decreases at each cutting operation. Moreover, we have  $\varphi(\mathcal{S}) = 2g - k$ , and by the previous paragraph  $\varphi(\mathcal{S}')$  is at least  $-(k + g + b)$ . This proves (1).

By Lemma 7.5.2, for any surface of genus  $g$  without boundary, there are at most  $g + 3$  ways of performing a cutting operation up to homeomorphism. After a sequence of essential cutting operations, we have a surface  $\mathcal{S}'$  with at most  $k + g + b$  connected components, with total genus at most  $g$ , and with  $b$  boundary components. The number

of surfaces that can be obtained from  $\mathcal{S}'$  by a cutting operation is at most  $k(g+3)(b+1)$ , since we first choose which connected component to cut along, the way to cut it ignoring the boundary components, and the number of boundary components in each connected component (if the cut is separating).  $\square$

We can now conclude the proof of this section.

*Proof of Proposition 7.5.1.* First, if the detached surface of  $\mathcal{C}$  is non-empty, then we test the planarity of each connected component of  $G$  [HT74] in  $O(n)$  time, and remove every connected component of  $G$  that is planar; obviously, this does not affect the embeddability of  $G$  on  $\mathcal{C}$ .

In a second step, we split each isolated segment of  $\mathcal{C}$  into five isolated segments. For each subset  $S$  of the isolated segments, we build a new 2-complex obtained from  $\mathcal{C}$  by removing  $S$ . We obtain  $2^{O(c)}$  2-complexes, each of size  $O(c)$ . The input graph  $G$  embeds on  $\mathcal{C}$  if and only if it embeds into one of these 2-complexes; moreover, if  $G$  embeds on  $\mathcal{C}$ , it embeds into one of the 2-complexes in a way that every isolated segment is covered by the embedding. (Indeed, remember that  $G$  has at most one connected component homeomorphic to a segment.)

We now iteratively dissolve every degree-two vertex of  $G$ , and then subdivide  $5c$  times each edge of  $G$ . This new graph  $G'$  has at most  $5cn$  vertices and edges, and branchwidth at most  $w$ . Clearly,  $G$  embeds on  $\mathcal{C}$  if and only if  $G'$  embeds in one of the 2-complexes defined in the previous paragraph; moreover, if  $G$  embeds on  $\mathcal{C}$ , then  $G'$  has an embedding on one of these 2-complexes in which the relative interior of every edge is distinct from any singular point (and, as above, such that every isolated segment is covered by the embedding).

Each singular point  $p$  of each of these 2-complexes is incident to at least two link components. For each such singular point  $p$  and for each partition of the link components at  $p$ , we replace  $p$  with new vertices, one for each element in the partition; two link components at  $p$  stay adjacent via one of these new vertices if and only if these link components are in the same part. We obtain  $c^{O(c)}$  2-complexes, each of size  $O(c)$ . The input graph  $G$  embeds on  $\mathcal{C}$  if and only if  $G'$  embeds in one of these 2-complexes; moreover, if  $G$  embeds on  $\mathcal{C}$ , then  $G'$  has an embedding into one of these 2-complexes in which every link component of each singular point  $p$  is used by an edge of  $G$  connected to  $p$  in that link component (and, as above, such that every isolated segment is covered by the embedding, and such that the relative interior of every edge is distinct from any singular point).

Every embedding of a graph into a 2-complex can be perturbed so that it avoids the boundary of the 2-complex, except possibly at singular points. This means that, if  $G$  embeds on  $\mathcal{C}$ , then  $G'$  has a proper cellular embedding into one of the 2-complexes built in the previous paragraph, except that the faces of  $G'$  may fail to be disks, but are more general (connected) surfaces with boundary.

To dispense ourselves from this latter exception, we need to build more 2-complexes. This case occurs only if the detached surface is non-empty, so by our earlier preprocessing, we can assume that  $G'$  contains no planar connected component, and so has  $O(c)$



connected components (because, by Lemma 7.2.4, in order for  $G'$  to be embeddable on a 2-complex of size  $O(c)$ , it must have genus  $O(c)$ ).

The detached surface  $\mathcal{S}$  is a surface (possibly disconnected, possibly with boundary); the trace of the set of singular points on  $\mathcal{S}$  corresponds to marked points, some in the interior of  $\mathcal{S}$ , some on the boundary. Henceforth, we regard the former as small boundary components. For each 2-complex obtained above, we consider, up to homeomorphism, all 2-complexes arising from zero, one, or several essential cutting operations on  $\mathcal{S}$ , and then by removing an arbitrary subset of the connected components of the resulting surface. Up to homeomorphism, by Lemma 7.5.3, there are  $c^{O(c)}$  ways of cutting  $\mathcal{S}$ ; since we consider all 2-complexes obtained up to homeomorphism, we need to consider each boundary component of the detached surface  $\mathcal{S}$  as labelled (which is not the case in Lemma 7.5.3); however, this only adds a factor of  $c^{O(c)}$ . In total, we obtain, in  $c^{O(c)}$  time,  $c^{O(c)}$  2-complexes, each of size  $O(c)$ . Then, for each such 2-complex, we consider all possible ways of removing an arbitrary subset of connected components of the 2-complex; the number of the resulting 2-complexes is still  $c^{O(c)}$ . By construction, the input graph  $G$  embeds on  $\mathcal{C}$  if and only if  $G'$  embeds in one of these 2-complexes. Moreover, assume that it is the case; as shown above,  $G'$  has a proper embedding into one of the 2-complexes defined in the previous paragraph, except that faces of  $G'$  are (connected) surfaces, not necessarily disks. Whenever a face has non-empty boundary and is not homeomorphic to a disk, we perform an essential cutting operation of that face along a closed curve inside that face; the closed curve along which we cut is non-contractible in  $\mathcal{S}$ , because otherwise it would bound a disk in  $\mathcal{S}$ , which would itself contain a planar connected component of  $G'$ , and we have shown above that we may assume that no such component exists. After iterating this operation as much as possible, every face of  $G'$  in the resulting 2-complex is either is a disk or has empty boundary; in the latter case,  $G'$  avoids the corresponding connected component, so we can simply remove it. Eventually, after a number of essential cutting operations of  $\mathcal{S}$  and removing some connected components of the 2-complex, the embedding of  $G'$  is cellular in one of the  $c^{O(c)}$  2-complexes of size  $O(c)$  enumerated above.  $\square$

## 7.6 Algorithm for bounded branchwidth: Proof of Theorem 7.1.2

*Proof of Theorem 7.1.2.* By Proposition 7.2.1, we can assume that  $\mathcal{C}$  has no 3-book and no connected component that is reduced to a single vertex, and that  $G$  has no connected component reduced to a single vertex and at most one connected component homeomorphic to a segment. If necessary, we convert the combinatorial description of  $\mathcal{C}$  into the topological data structure (Section 7.2.3).

We apply Proposition 7.5.1. In  $c^{O(c)} + O(cn)$  time, we obtain a graph  $G'$  and a set of  $c^{O(c)}$  2-complexes  $\mathcal{C}_i$  such that:

1. each  $\mathcal{C}_i$  and  $G'$  satisfy the properties of Proposition 7.2.1;
2.  $G'$  has at most  $5cn$  vertices and  $5cn$  edges, and branchwidth at most  $w$ ;

3. each  $\mathcal{C}_i$  has size at most  $c$ ;
4. if, for some  $i$ ,  $G'$  embeds into  $\mathcal{C}_i$ , then  $G$  embeds into  $\mathcal{C}$ ;
5. if  $G$  embeds into  $\mathcal{C}$ , then for some  $i$ ,  $G'$  has a proper cellular embedding into  $\mathcal{C}_i$ .

We then run the algorithm from Proposition 7.4.1 in each of the instances  $(\mathcal{C}_i, G')$ , in total time  $(c + w)^{O(c+w)}n$ . This algorithm correctly reports either that  $G'$  has no sparse proper cellular embedding into  $\mathcal{C}_i$  or that  $G'$  has an embedding into  $\mathcal{C}_i$ . If for at least one of these instances, the algorithm reports that  $G'$  embeds into  $\mathcal{C}_i$ , then we report that  $G$  embeds into  $\mathcal{C}$ . Otherwise, we report that  $G$  does not embed into  $\mathcal{C}$ .

There remains to prove that the algorithm is correct. If our algorithm reports that  $G$  embeds into  $\mathcal{C}$ , then it is obviously indeed the case (Property (4) above). Conversely, let us assume that  $G$  has an embedding into  $\mathcal{C}$ . Thus, by Property (5) above, let  $i$  be such that  $G'$  has a proper cellular embedding into  $\mathcal{C}_i$ . By Proposition 7.3.1,  $G'$  also has such an embedding into  $\mathcal{C}_i$  that is sparse. Thus, the algorithm in Proposition 7.4.1 (correctly) reports that  $G'$  has an embedding into  $\mathcal{C}_i$ , and finally our overall algorithm reports that  $G$  has an embedding into  $\mathcal{C}$ .  $\square$

## 7.7 Reduction to bounded branchwidth: Proof of Theorem 7.1.3

This section is devoted to the proof of Theorem 7.1.3. The deterministic and randomized algorithms differ only in the algorithm that we use as a subroutine to compute a large grid minor. The proof technique is based on an irrelevant vertex method; we borrow ingredients to Kociumaka and Pilipczuk [KP19, Section 5], but some new arguments are needed, in particular in the beginning of the proof of Proposition 7.7.1.

### 7.7.1 Finding a large planar part

As an intermediate goal towards the proof of Theorem 7.1.3, we will prove in this subsection:

**Proposition 7.7.1.** *Let  $G$  be a graph with  $n$  vertices and edges and  $g \geq 2$  be an integer. We can do one of the following:*

1. compute a rooted branch decomposition of  $G$  of width  $g^{O(1)}$ ;
2. correctly report that  $G$  has genus at least  $g$ ;
3. or compute a cycle  $\gamma$  of  $G$  such that one connected component of  $G - \gamma$  is planar and contains a subdivision of the  $(g \times g)$ -wall, which is also computed.

*either in deterministic  $f(c) \cdot n^2$  time for some computable function  $f$ , or in expected polynomial time.*

We will use the following lemma.

**Lemma 7.7.2.** *Let  $G$  be a graph with  $n$  vertices and edges and  $k \geq 2$ . Then, one can compute either a rooted branch decomposition of  $G$  of width  $k^{O(1)}$  or a  $(k \times k)$ -grid minor of  $G$  in deterministic  $f(c) \cdot n^2$  time for some computable function  $f$ , or in expected polynomial time.*

*Proof.* We can obviously assume that  $k$  is at most the number of vertices of  $G$ , for otherwise any rooted branch decomposition of  $G$  has width at most  $k$ . Let  $d$  be a large enough universal constant, which can be computed from the hidden constants in the  $O(\cdot)$  notation of the results used below.

We first use any algorithm to approximate the treewidth, e.g., Fomin et al. [Fom+18, Theorem 1.1]: Provided  $d$  is large enough, in polynomial time, we either compute a tree decomposition of width at most  $(dk^d)^2$ , and thus immediately obtain a (rooted) branch decomposition of width  $(dk^d)^2$  [RS91, Theorem 5.1], as desired, or correctly report that the treewidth is at least  $dk^d$ .

In the latter case, provided  $d$  is large enough,  $G$  has a  $(k \times k)$ -grid minor, by a result by Chekuri and Chuzhoy [CC16]. We can compute it in expected polynomial time, by an algorithm from the same article [CC16]. Alternatively, we can compute it in deterministic  $f(k) \cdot n^2$  time, for some computable function  $f$ , by a result by Robertson and Seymour [RS95, Algorithm 4.4].  $\square$

We remark that the above proof contains some bottlenecks in the running time in Theorems [7.1.3] and [7.1.1]. Specifically:

- the randomness is solely due to the above use of the algorithm in the article by Chekuri and Chuzhoy [CC16];
- the fact that the dependence in the size of the 2-complex is not specified is solely due to the use of Robertson and Seymour [RS95, Algorithm 4.4] in computing a grid minor, but it can in principle be made explicit.

*Proof of Proposition [7.7.1].* We can again obviously assume that  $g$  is at most the number of vertices of  $G$ . We apply Lemma [7.7.2] with  $k = 60\lceil\sqrt{g}\rceil^4$ . If the outcome is a rooted branch decomposition, then the algorithm returns it (Case 1). Otherwise, we have computed a  $(60\lceil\sqrt{g}\rceil^4 \times 60\lceil\sqrt{g}\rceil^4)$ -grid minor of  $G$ , and thus a subgraph  $\tilde{W}$  of  $G$  that is a subdivision of a  $(60\lceil\sqrt{g}\rceil^4 \times 60\lceil\sqrt{g}\rceil^4)$ -wall  $W$ .

We first compute disjoint non-adjacent  $(50g\lceil\sqrt{g}\rceil \times 50g\lceil\sqrt{g}\rceil)$ -walls  $W_1, \dots, W_g$  of  $W$ , and the corresponding subdivisions  $\tilde{W}_1, \dots, \tilde{W}_g$  that are subgraphs of  $\tilde{W}$ , in a way that  $W - W_i$  is connected for each  $i$ . For each  $i$ , we consider the subgraph  $G_i$  of  $G$  induced by the vertices  $v$  of  $G$  such that: (1) there exists a path from  $v$  to  $\tilde{W}_i$ ; (2) every path from  $v$  to  $\tilde{W}_j$ , for some  $j \neq i$ , uses at least one vertex from  $\tilde{W}_i$ . The graphs  $G_i$  are pairwise disjoint. We test the planarity of each of them in linear time [HT74]. If all of them are non-planar, then  $G$  has genus at least  $g$ , so we correctly report this (Case 2). So without loss of generality, one of these graphs, say  $G_1$ , is planar, and our algorithm computes it.

By 3-connectivity,  $W_1$ , and thus also  $\tilde{W}_1$ , has a unique combinatorial embedding in the plane, up to symmetry and up to the choice of the outer (infinite) face; we consider

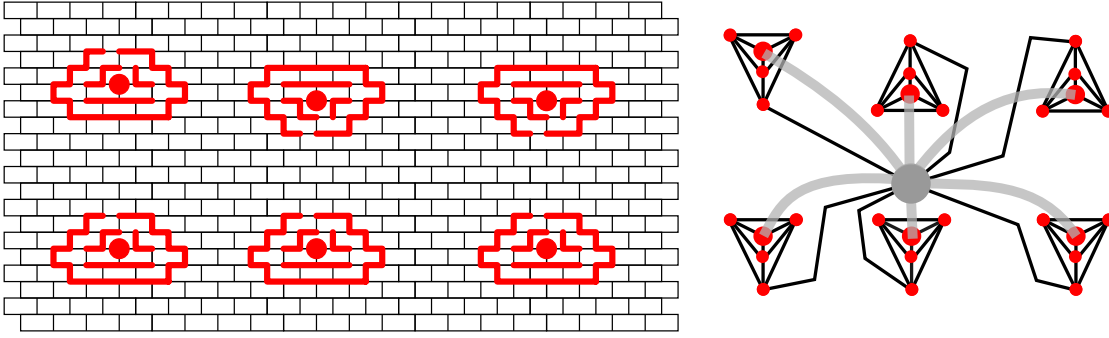


Figure 7.8: Illustration of the proof of the claim in Proposition 7.7.1. The wall  $W_1$  (left), and a schematic view of the  $J_g$  minor (right), illustrated with  $g = 6$ . Let  $K_5^-$  be the graph  $K_5$  with one edge removed. If  $W_1$  has many vertices connected to the outside, then a set  $U$  of at least  $g$  vertices connected to the outside (represented as big disks on the left) are pairwise distant. In the neighborhood of these vertices, we build a  $K_5^-$  minor for each vertex in  $U$ , in which the vertex of  $U$  is the “central” vertex (the thick paths need to be contracted to obtain copies of  $K_5^-$ ). Using the fact that the neighborhoods are far apart, we build a  $J_g$ -minor of  $G$  (right), in which the apex results from the contraction of the subdivided wall  $\dot{W}_1$  minus the union of the  $K_5^-$  minors, and each path connecting  $u \in U$  to  $\dot{W} - \dot{W}_1$  (which is connected by assumption) is contracted, except its first edge, to connect  $u$  to the apex.

the natural embedding of  $W_1$  in which the outer face has the largest degree. An *inner vertex* of  $W_1$  is one that is at distance at least 6 from the outer face in the natural embedding of  $W_1$ . Remark that each vertex of  $W_1$  is a vertex of  $\dot{W}_1$ . We say that a vertex  $u$  of  $W_1$  is *connected to the outside* if there is, in  $G$ , a path whose vertices are, in this order,  $u$ , possibly some vertices of  $\dot{W}_1$  but not of  $W_1$ , possibly some vertices of  $G - \dot{W}_1$ , and finally one vertex in  $\dot{W} - \dot{W}_1$ .

We claim the following: *If at least  $1000g$  inner vertices of  $W_1$  are connected to the outside, then  $G$  has genus at least  $g$ .* The strategy is similar to the argument in Kociumaka and Pilipczuk [KP19, Lemma 5.3]; we summarize the proof. If at least  $1000g$  inner vertices of  $W_1$  are connected to the outside, then a set  $U$  of  $g$  inner vertices of  $W_1$  are connected to the outside, and at pairwise distance at least 16 in  $W_1$ . This implies that  $G$  contains, as a minor, the graph  $J_g$  obtained from  $g$  copies of  $K_5$  by subdividing an edge from each copy with a degree-two vertex and identifying these  $g$  new vertices into a single vertex, the *apex* of  $J_g$ ; see Figure 7.8. This graph has genus at least  $g$ , by the main result of an article by Miller [Mil87, Theorem 1]. This proves the claim.

In polynomial time, we can compute the inner vertices of  $W_1$  connected to the outside. If there are at least  $1000g$  of these, we report that  $G$  has genus at least  $g$  (Case 2), which is correct by the above claim. Otherwise, from the  $(50g\lceil\sqrt{g}\rceil \times 50g\lceil\sqrt{g}\rceil)$ -wall  $W_1$ , we can compute a cycle  $\gamma$  in  $W_1$ , enclosing a  $(g \times g)$ -wall  $W'_1$  in  $W_1$  (in the natural embedding of  $W_1$ ) such that no vertex of  $W_1$  inside  $\gamma$  (in the natural embedding of  $W_1$ ) is connected to the outside. Let  $\dot{\gamma}$  be the cycle of  $\dot{W}_1$  corresponding to  $\gamma$ , and let  $H$  be the connected

component of  $G - \dot{\gamma}$  containing the vertices of this  $(g \times g)$ -wall. There remains to prove that  $H$  is planar, and since  $G_1$  is planar, it suffices to prove that  $H$  is a subgraph of  $G_1$ . If it were not the case,  $H$  would contain a vertex of  $\dot{W}_j$  for some  $j \neq 1$ ; but that would imply that a vertex of  $W'_1$  is connected to the outside, which is not the case. We can thus correctly report  $\gamma$  (Case 3).  $\square$

### 7.7.2 Finding an irrelevant vertex

The following proposition will imply that if the third possibility in the statement of Proposition 7.7.1 holds (for some  $g$  large enough), then one has an *irrelevant vertex* for the embedding instance.

**Proposition 7.7.3.** *Let  $\mathcal{C}$  be a 2-complex with  $c \geq 1$  simplices. Let  $G$  be a graph and  $\gamma$  be a cycle in  $G$  such that one connected component of  $G - \gamma$  is planar and contains a subdivision of the  $(1000c \times 1000c)$ -wall. Let  $v$  be the central vertex of this wall. Then  $G$  is embeddable on  $\mathcal{C}$  if and only if  $G - v$  is.*

The proof of Proposition 7.7.3 builds upon the article by Kociumaka and Pilipczuk [KP19, Section 5.3]: The strategy is the same, up to simple variations to take into account the fact that we consider a 2-complex, not a surface. Rather than repeating the same arguments, we summarize the proof, emphasizing the differences.

A *circular wall*  $W$  [KP19, Figure 8] of height  $h$  and circumference  $\ell$  is a 3-regular graph that, in some embedding of  $W$  in the plane, is represented as the union of  $h$  vertex-disjoint cycles, called *circles*, organized in a concentric way, such that any two consecutive circles are connected by  $\ell$  *radial edges*; in successive layers, the radial edges are interleaved. A *ring*  $R$  of this circular wall  $W$  is a subgraph contained in the (closed) annulus between two circles at distance four in the radial order; a ring thus contains five vertex-disjoint circles, consecutive in this concentric order, together with edges connecting them; the *central circle* of  $R$  is the third one in that order. A *central brick* of  $R$  is the boundary of a face of  $W$  incident to its central circle.

If  $G$  is embeddable on  $\mathcal{C}$ , then obviously  $G - v$  is also embeddable on  $\mathcal{C}$ ; the hard part is the reverse direction. In the given subdivision of the  $(1000c \times 1000c)$ -wall in  $G$ , we first compute a subdivision  $\dot{W}$  of a circular wall  $W$  of height  $45c + 10$  and circumference three, so that  $v$  is located inside the inner circle of this circular wall in the planar embedding of  $W$ ; see Kociumaka and Pilipczuk [KP19, Figures 8 and 9].

Let us consider an embedding  $\psi$  of  $G - v$  on  $\mathcal{C}$ . This induces embeddings of  $\dot{W} - v$  and  $W - v$  on  $\mathcal{C}$ , also denoted by  $\psi$ . Each ring of  $W$  corresponds naturally to a subdivided ring of  $\dot{W}$ . Such a subdivided ring is *embedded plainly* in  $\psi$  if it lies in the detached surface  $\mathcal{S}$  of  $\mathcal{C}$  and each central brick of the ring is a two-sided closed curve in  $\mathcal{S}$  and bounds a disk on  $\mathcal{S}$ , the interior of which does not contain any vertex or edge of  $\dot{W}$ .

We have the following lemma.

**Lemma 7.7.4.** *Some subdivided ring of  $\dot{W}$  is embedded plainly in  $\psi$ .*

*Proof.* At most  $c$  vertices or edges of  $W$  are mapped, under  $\psi$ , to a singular point of  $\mathcal{C}$ . At most  $c$  edges of  $W$  are mapped, under  $\psi$ , to an isolated segment of  $\mathcal{C}$ . Our

circular wall of height  $5(9c + 2)$  and circumference 3 contains  $9c + 2$  vertex-disjoint rings of circumference 3. Thus, there exists a set  $\mathcal{R}$  of  $7c + 2$  vertex-disjoint rings of  $W$  of circumference 3, the vertices and edges of which are mapped, under  $\psi$ , to the complement of the singular points and the isolated segments of  $\mathcal{C}$ .

In other words,  $\psi$  maps each ring of  $\mathcal{R}$  to the detached surface  $\mathcal{S}$  of  $\mathcal{C}$ , and thus also to the surface  $\bar{\mathcal{S}}$  obtained by attaching a handle to every boundary component of  $\mathcal{S}$ . The resulting surface has a natural cellular graph embedding with at most  $7c$  edges, and thus has genus at most  $7c$ . Since  $\mathcal{R}$  contains  $7c + 2$  rings, this implies that some ring in  $\mathcal{R}$  is embedded plainly, by  $\psi$ , in  $\bar{\mathcal{S}}$ ; see the proof of Kociumaka and Pilipczuk [KP19, Lemma 5.4] for details. Hence it is also embedded plainly, by  $\psi$ , in  $\mathcal{C}$ .  $\square$

The rest of the proof of Proposition 7.7.3 uses the same arguments as in Kociumaka and Pilipczuk [KP19], so we only summarize it:

*Proof of Proposition 7.7.3.* Let  $R$  be the ring obtained from Lemma 7.7.4. Its central cycle  $\gamma$  is mapped, under  $\psi$ , to a two-sided cycle in the detached surface  $\mathcal{S}$ , and moreover, the edges of  $R$  incident to  $\gamma$  are partitioned, under  $\psi$ , between the sides of  $\gamma$  exactly as in the natural embedding [KP19, Corollary 5.5]. Since  $R$  is embedded plainly in the detached surface, one can embed the (planar) part of  $G$  that lies inside  $R$  in a neighborhood of  $\gamma$ , and then insert  $v$  without interfering with the rest of the embedding.  $\square$

### 7.7.3 Proof of Theorem 7.1.3

*Proof of Theorem 7.1.3.* We first apply Proposition 7.2.1: without loss of generality,  $\mathcal{C}$  has no 3-book and no connected component that is reduced to a single vertex, and  $G$  has no connected component reduced to a single vertex, and at most one connected component homeomorphic to a segment. Let  $n$  be the number of vertices and edges of the input graph  $G$ , and  $c$  be the number of simplices of  $\mathcal{C}$ . We apply Proposition 7.7.1 to the graph  $G$ , letting  $g = 1000c$ . In deterministic  $f(c) \cdot n^2$  time for some computable function  $f$ , or in expected time polynomial in  $n$  and  $c$ , we obtain one of the following outcomes:

1. a rooted branch decomposition of  $G$  of width  $O(g)^{O(1)}$ ;
2. that  $G$  has genus at least  $g$ , and is thus not embeddable on  $\mathcal{C}$  (Lemma 7.2.4);
3. a cycle  $\gamma$  of  $G$  such that one connected component of  $G - \gamma$  is planar and contains a subdivision of the  $(g \times g)$ -wall.

In the first two cases, we are done. In the third case, by applying Proposition 7.7.3, we obtain a vertex  $v$  such that  $G$  embeds on  $\mathcal{C}$  if and only if  $G - v$  does. By iterating the same procedure a number of times that is at most the number of vertices of  $G$ , we necessarily reach case (1) or (2), which concludes.  $\square$

We remark that the proof goes through if the input 2-complex is given in the form of the topological data structure, and  $c$  denotes its size, instead of the number of simplices of  $\mathcal{C}$ .

As mentioned above, the proof of Theorem [7.1.1](#) follows immediately from Theorems [7.1.2](#) and [7.1.3](#).

## Conclusion

In this thesis, we studied the embeddability of graphs on 2-complexes which is obviously **NP**-hard as a generalization of embedding of graphs on surfaces. We presented two algorithms solving this problem, the first also proving that the problem is **NP**-complete and the second being FPT with parameter the size on the input 2-complex. Now, a FPT algorithm is the best we can reasonably ask for a **NP**-complete problem from general complexity class point of view. Nonetheless, there are still many natural problems that arise from it.

**A more efficient algorithm to solve EMBED** In this thesis, we presented an algorithm deciding whether a graph is embeddable on a 2-complex in time  $f_0(c)n^3$  for some computable function  $f_0$ . Nonetheless, the more restrictive problem when the target space is a surface is known to have algorithms linear in the size of the input graph, solving it in time  $f_1(g)n$  for some function  $f_1$ , where  $g$  is the genus of the input surface and  $n$  is the size of the input graph [Moh99; KMR08]. There is no known obstruction to find an algorithm that solves EMBED in similar time, linear in the size of the graph. Thus, it could be reasonable to look for an algorithm deciding the embeddability of graphs on 2-complexes in time  $O(f_2(c)n)$  for some function  $f_2$ .

**Looking for specific topological embedding** For the problem that we have considered in this thesis, we have completely ignored the combinatorial structure of the 2-complex into which we try to embed the input graph: any two homeomorphic complexes are equivalent for this problem

In this thesis, we considered the problem where we looked for *any* topological embedding of the input graph on the input complex. It would be very interesting to look only for embeddings having some specific property. In order to still consider embeddings up to homeomorphism, the properties we consider must translate through homeomorphism. For example, we could consider a LEW-embedding of a graph  $G$  defined as a topological embedding of  $G$  such that there is no non-contractible cycle in the image of  $G$  with fewer edges than the facial walk of any face of the embedding. Then, we consider the following



problem that extends a version on orientable surfaces studied by C.Thomassen [Tho90]: given a graph  $G$  and a 2-complex  $\mathcal{C}$  such that  $G$  is embeddable on  $\mathcal{C}$ , decide whether there exists LEW-embedding of  $G$  on  $\mathcal{C}$ . As shown by C.Thomassen [Tho90], these Lew-embeddings restricted to orientable surfaces have some strong properties. First, if a graph  $G$  have a LEW-embedding on an orientable surface of genus  $g$ , then the orientable genus of  $G$  is  $g$ . Moreover, a 3-connected graph  $G$  have at most one LEW-embedding on an orientable surface.

It is reasonable to hope that we could obtain some similar properties for LEW-embeddings on 2-complexes.

**Not looking for topological embeddings** For the problem that we have considered in this thesis, we have completely ignored the combinatorial or geometric structure of the 2-complex into which we try to embed the input graph: any two homeomorphic complexes are equivalent for this problem. This makes impossible to consider any potential property of an embedding that are not maintained through homeomorphisms. Then, by considering the exact embedding, it is possible to look for some specific embeddings. For example, on a 2-complex  $\mathcal{C}$  with a metric, we can define a straight line embedding of a graph as an embedding where all edges are mapped as a segment (instead of an arbitrary curve for general embeddings) between the image of its endpoints. This way, we could define the problem of the straight line embeddability problem: given a graph  $G$  and a 2-complex  $\mathcal{C}$  with a metric such that there exists a topological embedding of  $G$  on  $\mathcal{C}$ , decide whether there exists a straight line embedding of  $G$  on  $\mathcal{C}$ . Similarly, we could look for embeddings having some property using the exact combinatorial structure of the input 2-complex. For example, we could consider the "insimplex" embeddability problem: given a graph  $G$  and a 2-complex  $\mathcal{C}$  such that there exists a topological embedding of  $G$  on  $\mathcal{C}$ , decide whether there exists an embedding of  $G$  on  $\mathcal{C}$  such that the image of each edge is included in either one edge or one triangle of  $\mathcal{C}$ .

**Finding some other FPT parameters** In this thesis, we presented an FPT algorithm deciding the embeddability of graphs on 2-complexes with parameter the size of the input complex. Similarly, as presented in Section 4.1.2, there exists some FPT algorithms solving the embeddability of graphs on surfaces where the parameter is the genus of the input surface. For the embeddability of graphs on surfaces there exists some other parameters such that there is an FPT algorithm deciding it (for example the treewidth of the input graph [Bod88]). Similarly, we could try to find a parameter  $p$  other than the size of the input complex such that there exists a FPT algorithm deciding whether a graph is embeddable on a 2-complex with parameter  $p$ . Similarly to surfaces, a great candidate for  $p$  could also be the treewidth of the input graph.

**Finding characteristics of a 2-complex characterizing the set of graphs embeddable on it** We can define an equivalence relation  $\mathcal{R}$  on 2-complexes by: two 2-complexes are equivalent if and only the set of graph embeddable on one is the same

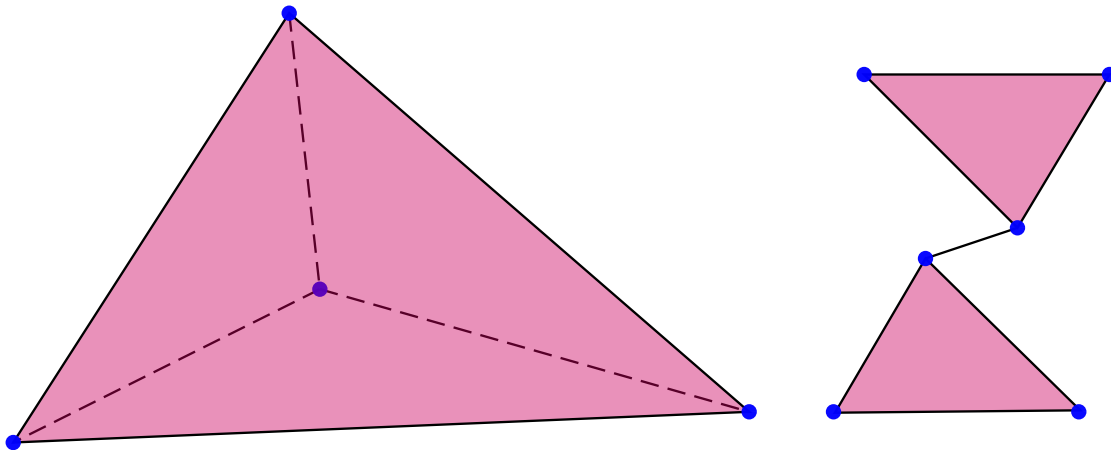


Figure 8.1: Two complexes, the sphere and two triangles linked by an isolated edge, that are not homeomorphic while having the same set of graphs embeddable on it.

as the set of graphs embeddable on the other.  $\mathcal{R}$  is a coarser relation than being homeomorphic since two homeomorphic 2-complexes obviously have the same set of graphs embeddable on it while there is some couples of 2-complexes equivalent with  $\mathcal{R}$  that are not homeomorphic (See Figure 8.1 for an example of such couple of 2-complexes). It would be interesting to find whether it is possible to decide efficiently the relation  $\mathcal{R}$ .

Moreover, as presented in Section 3.4.3, two connected surfaces are equivalent with  $\mathcal{R}$  if and only if the two have the same genus and are either both orientable or both non-orientable. Thus, the equivalence class of  $\mathcal{R}$  to which belongs a connected surface is characterized by its genus and orientability. Then, similarly, it would be very interesting to find a simple combinatorial characterization of the equivalent class of  $\mathcal{R}$  to which belongs a 2-complex.



# Bibliography

- [AGK08] Isolde Adler, Martin Grohe, and Stephan Kreutzer. “Computing excluded minors”. In: *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2008, pp. 641–650.
- [AH76] Kenneth Appel and Wolfgang Haken. “Every planar map is four colorable”. In: *Bulletin of the American mathematical Society* 82.5 (1976).
- [ALS91] Stefan Arnborg, Jens Lagergren, and Detlef Seese. “Easy problems for tree-decomposable graphs”. In: *Journal of Algorithms* 12.2 (1991), pp. 308–340.
- [BA01] Hans L Bodlaender and Babette van Antwerpen-de Fluiter. “Reduction algorithms for graphs of small treewidth”. In: *Information and Computation* 167.2 (2001), pp. 86–119.
- [BM07] Denis Blackmore and Yuriy Mileyko. “Computational differential topology”. In: *Applied general topology* 8.1 (2007), pp. 35–92.
- [BM88] Daniel Bienstock and Clyde L Monma. “On the complexity of covering vertices by faces in a planar graph”. In: *SIAM Journal on Computing* 17.1 (1988), pp. 53–76.
- [Bod+16] Hans L Bodlaender, Pål Grønås Drange, Markus S Dregi, Fedor V Fomin, Daniel Lokshtanov, and Michał Pilipczuk. “A  $c^k n^5$ -Approximation Algorithm for Treewidth”. In: *SIAM Journal on Computing* 45.2 (2016), pp. 317–378.
- [Bod88] Hans L Bodlaender. “Dynamic programming on graphs with bounded treewidth”. In: *International Colloquium on Automata, Languages, and Programming*. Springer. 1988, pp. 105–118.
- [Bod96] Hans L Bodlaender. “A linear-time algorithm for finding tree-decompositions of small treewidth”. In: *SIAM Journal on computing* 25.6 (1996), pp. 1305–1317.
- [Boo59] William W Boone. “The word problem”. In: *Annals of mathematics* (1959), pp. 207–265.
- [Bri62] JL Britton. “William W. Boone. The word problem. *Annals of mathematics*, vol. 70 (1959), pp. 207–265.” In: *The Journal of Symbolic Logic* 27.2 (1962), pp. 238–241.

- [BT97] Hans L Bodlaender and Dimitrios M Thilikos. “Constructive linear time algorithms for branchwidth”. In: (1997), pp. 627–637.
- [Bur+19] Benjamin Burton, Sergio Cabello, Stefan Kratsch, and William Pettersson. “The parameterized complexity of finding a 2-sphere in a simplicial complex”. In: *SIAM Journal on Discrete Mathematics* 33.4 (2019), pp. 2092–2110.
- [CC16] Chandra Chekuri and Julia Chuzhoy. “Polynomial bounds for the grid-minor theorem”. In: *Journal of the ACM* 63.5 (2016), Article 40.
- [Cha+08] Erin W. Chambers, Éric Colin de Verdière, Jeff Erickson, Francis Lazarus, and Kim Whittlesey. “Splitting (complicated) surfaces is hard”. In: *Computational Geometry: Theory and Applications* 41.1–2 (2008), pp. 94–110.
- [Che+06] Jianer Chen, Xiuzhen Huang, Iyad A Kanj, and Ge Xia. “Strong computational lower bounds via parameterized complexity”. In: *Journal of Computer and System Sciences* 72.8 (2006), pp. 1346–1367.
- [Chu15] Julia Chuzhoy. “Excluded grid theorem: Improved and simplified”. In: *Proceedings of the forty-seventh annual ACM symposium on Theory of Computing*. 2015, pp. 645–654.
- [ČKV17] Martin Čadek, Marek Krčál, and Lukáš Vokřínek. “Algorithmic solvability of the lifting-extension problem”. In: *Discrete & Computational Geometry* 57.4 (2017), pp. 915–965.
- [CM14] Éric Colin de Verdière and Arnaud de Mesmay. “Testing graph isotopy on surfaces”. In: *Discrete & Computational Geometry* 51.1 (2014), pp. 171–206.
- [CM21] Éric Colin de Verdière and Thomas Magnard. “An FPT algorithm for the embeddability of graphs into two-dimensional simplicial complexes”. 29th Annual European Symposium on Algorithms (ESA 2021), to appear. 2021.
- [CMM18] Éric Colin de Verdière, Thomas Magnard, and Bojan Mohar. “Embedding Graphs into Two-Dimensional Simplicial Complexes”. In: *34th International Symposium on Computational Geometry (SoCG 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. 2018. Full version in arXiv:<https://arxiv.org/abs/1803.07032>.
- [Col18] Éric Colin de Verdière. “Computational topology of graphs on surfaces”. In: *Handbook of Discrete and Computational Geometry*. Ed. by Jacob E. Goodman, Joseph O’Rourke, and Csaba Toth. third. See <http://www.arxiv.org/abs/1702.05358>. CRC Press LLC, 2018. Chap. 23.
- [Cou90] Bruno Courcelle. “The monadic second-order logic of graphs. I. Recognizable sets of finite graphs”. In: *Information and computation* 85.1 (1990), pp. 12–75.
- [Cyg+15] Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized algorithms*. Vol. 4. 8. Springer, 2015.

- [DEG99] Tamal K Dey, Herbert Edelsbrunner, and Sumanta Guha. “Computational topology”. In: *Contemporary mathematics* 223 (1999), pp. 109–144.
- [DH05] Erik D. Demaine and MohammadTaghi Hajiaghayi. “Graphs excluding a fixed minor have grids as large as treewidth, with combinatorial and algorithmic applications through bidimensionality”. In: *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2005, pp. 682–689.
- [Dud13] H Dudeney. “Perplexities, with some easy puzzles for beginners”. In: *The Strand Magazine* (1913), p. 110.
- [DW04] Vida Dujmovic and David R Wood. “On linear layouts of graphs”. In: *Discrete Mathematics and Theoretical Computer Science* 6.2 (2004).
- [DWW00] Colm Ó Dúnlaing, Colum Watt, and David Wilkins. “Homeomorphism of 2-complexes is equivalent to graph isomorphism”. In: *International Journal of Computational Geometry & Applications* 10.05 (2000), pp. 453–476.
- [EH10] Herbert Edelsbrunner and John Harer. *Computational topology: an introduction*. American Mathematical Soc., 2010.
- [EN11] Jeff Erickson and Amir Nayyeri. “Shortest non-crossing walks in the plane”. In: *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*. SIAM. 2011, pp. 297–308.
- [Epp03] David Eppstein. “Dynamic generators of topologically embedded graphs”. In: *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2003, pp. 599–608.
- [Fom+18] Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, Michał Pilipczuk, and Marcin Wrochna. “Fully polynomial-time parameterized computations for graphs and matrices of low treewidth”. In: *ACM Transactions on Algorithms* 14.3 (2018), Article 34.
- [FOR06] Hubert de Fraysseix, Patrice Ossona de Mendez, and Pierre Rosenstiehl. “Trémaux trees and planarity”. In: *International Journal of Foundations of Computer Science* 17.5 (2006), pp. 1017–1029.
- [FPP90] Hubert de Fraysseix, János Pach, and Richard Pollack. “How to draw a planar graph on a grid”. In: *Combinatorica* 10.1 (1990), pp. 41–51.
- [FWZ20] Marek Filakovsk, Uli Wagner, and Stephan Zhechev. “Embeddability of Simplicial Complexes is Undecidable”. In: *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM. 2020, pp. 767–785.
- [GH01] Joseph L Ganley and Lenwood S Heath. “The pagenumber of k-trees is  $O(k)$ ”. In: *Discrete Applied Mathematics* 109.3 (2001), pp. 215–221.
- [Gol+09] Petr A Golovach, Marcin Kamiński, Daniël Paulusma, and Dimitrios M Thilikos. “Induced packing of odd cycles in a planar graph”. In: *International Symposium on Algorithms and Computation*. Springer. 2009, pp. 514–523.

- [GOT18] Jacob E. Goodman, Joseph O’Rourke, and Csaba Tóth, eds. *Handbook of discrete and computational geometry*. third. Boca Raton, FL: CRC Press LLC, 2018.
- [GR81] Jonathan L Gross and Ronald H Rosen. “A combinatorial characterization of planar 2-complexes”. In: *Colloquium Mathematicum*. Vol. 44. 2. Institute of Mathematics Polish Academy of Sciences. 1981, pp. 241–247.
- [Hak73] Wolfgang Haken. “Connections between topological and group theoretical decision problems”. In: *Studies in Logic and the Foundations of Mathematics*. Vol. 71. Elsevier, 1973, pp. 427–441.
- [HT74] John Hopcroft and Robert Tarjan. “Efficient planarity testing”. In: *Journal of the ACM (JACM)* 21.4 (1974), pp. 549–568.
- [JM05] M Juvan and B Mohar. “Extending 2-restricted partial embeddings of graphs”. In: *Europ. J. Combin* 26 (2005), pp. 339–375.
- [JMM07] Martin Juvan, Jože Marinček, and Bojan Mohar. “Obstructions for simple embeddings”. In: *Australasian Journal of Combinatorics* 38 (2007), pp. 3–25.
- [JMM97] Martin Juvan, Jože Marinček, and Bojan Mohar. “Elimination of local bridges”. In: *Mathematica Slovaca* 47.1 (1997), pp. 85–92.
- [JT95] William Jaco and Jeffrey L Tollefson. “Algorithms for the complete decomposition of a closed 3-manifold”. In: *Illinois journal of mathematics* 39.3 (1995), pp. 358–406.
- [KKR12] Ken-ichi Kawarabayashi, Yusuke Kobayashi, and Bruce Reed. “The disjoint paths problem in quadratic time”. In: *Journal of Combinatorial Theory, Series B* 102 (2012), pp. 424–435.
- [KM07] Ken-ichi Kawarabayashi and Bojan Mohar. “Some recent progress and applications in graph minor theory”. In: *Graphs and combinatorics* 23.1 (2007), pp. 1–46.
- [KMR08] Ken-ichi Kawarabayashi, Bojan Mohar, and Bruce Reed. “A simpler linear time algorithm for embedding graphs into an arbitrary surface and the genus of graphs of bounded tree-width”. In: *Foundations of Computer Science, 2008. FOCS’08. IEEE 49th Annual IEEE Symposium on*. IEEE. 2008, pp. 771–780.
- [KMS13] Marek Krčál, Jiří Matoušek, and Francis Sergeraert. “Polynomial-time homology for simplicial Eilenberg–MacLane spaces”. In: *Foundations of computational mathematics* 13.6 (2013), pp. 935–963.
- [KP17] Tomasz Kociumaka and Marcin Pilipczuk. “Deleting vertices to graphs of bounded genus”. In: *Algorithmica* (2017), pp. 1–37.
- [KP19] Tomasz Kociumaka and Marcin Pilipczuk. “Deleting vertices to graphs of bounded genus”. In: *Algorithmica* 81 (2019), pp. 3655–3691.

- [KR07] Ken-ichi Kawarabayashi and Buce Reed. “Computing crossing number in linear time”. In: *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*. ACM. 2007, pp. 382–390.
- [KT10] Stephan Kreutzer and Siamak Tazari. “On brambles, grid-like minors, and parameterized intractability of monadic second-order logic”. In: *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*. SIAM. 2010, pp. 354–364.
- [Kur30] Casimir Kuratowski. “Sur le probleme des courbes gauches en topologie”. In: *Fundamenta mathematicae* 15.1 (1930), pp. 271–283.
- [Lin82] Sóstenes Lins. “Graph-encoded maps”. In: *Journal of Combinatorial Theory, Series B* 32 (1982), pp. 171–181.
- [Lov06] László Lovász. “Graph minor theory”. In: *Bulletin of the American Mathematical Society* 43.1 (2006), pp. 75–86.
- [LR12] Francis Lazarus and Julien Rivaud. “On the homotopy test on surfaces”. In: *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*. IEEE. 2012, pp. 440–449.
- [Mal94] Seth M Malitz. “Genus  $g$  graphs have pagenumber  $O(\sqrt{g})$ ”. In: *Journal of Algorithms* 17.1 (1994), pp. 85–109.
- [Mat+14] Jiří Matoušek, Eric Sedgwick, Martin Tancer, and Uli Wagner. “Embeddability in the 3-sphere is decidable”. In: *Proceedings of the thirtieth annual symposium on Computational geometry*. ACM. 2014, p. 78.
- [Mat+16] Jiří Matoušek, Eric Sedgwick, Martin Tancer, and Uli Wagner. “Untangling two systems of noncrossing curves”. In: *Israel Journal of Mathematics* 212 (2016), pp. 37–79.
- [Mat03] Jiří Matoušek. “The Borsuk–Ulam Theorem”. In: *Using the Borsuk–Ulam Theorem: Lectures on Topological Methods in Combinatorics and Geometry* (2003), pp. 21–46.
- [Mil87] Gary L Miller. “An additivity theorem for the genus of a graph”. In: *Journal of Combinatorial Theory, Series B* 43.1 (1987), pp. 25–47.
- [Moh06] Bojan Mohar. “Universal obstructions for embedding extension problems”. In: *Australasian Journal of Combinatorics* 36 (2006), p. 39.
- [Moh92] Bojan Mohar. “Combinatorial local planarity and the width of graph embeddings”. In: *Canadian Journal of Mathematics* 44.6 (1992), pp. 1272–1288.
- [Moh97] Bojan Mohar. “On the minimal genus of 2-complexes”. In: *Journal of Graph Theory* 24.3 (1997), pp. 281–290.
- [Moh99] Bojan Mohar. “A linear time algorithm for embedding graphs in an arbitrary surface”. In: *SIAM Journal on Discrete Mathematics* 12.1 (1999), pp. 6–26.
- [MT01] Bojan Mohar and Carsten Thomassen. *Graphs on surfaces*. Vol. 2. Johns Hopkins University Press Baltimore, 2001.



- [MTW09] Jiří Matoušek, Martin Tancer, and Uli Wagner. “Hardness of embedding simplicial complexes in  $\mathbb{R}^d$ ”. In: *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 2009, pp. 855–864.
- [Pat06] Maurizio Patrignani. “Planarity testing and embedding”. In: *Handbook of graph drawing and visualization*. Ed. by Roberto Tamassia. Chapman and Hall, 2006.
- [RS04] Neil Robertson and Paul D. Seymour. “Graph minors. XX. Wagner’s conjecture”. In: *Journal of Combinatorial Theory, Series B* 92 (2004), pp. 325–357.
- [RS86a] Neil Robertson and Paul D Seymour. “Graph minors. V. Excluding a planar graph”. In: *Journal of Combinatorial Theory, Series B* 41.1 (1986), pp. 92–114.
- [RS86b] Neil Robertson and Paul D. Seymour. “Graph minors. II. Algorithmic aspects of tree-width”. In: *Journal of algorithms* 7.3 (1986), pp. 309–322.
- [RS90] Neil Robertson and Paul D Seymour. “Graph minors. VIII. A Kuratowski theorem for general surfaces”. In: *Journal of Combinatorial Theory, Series B* 48.2 (1990), pp. 255–288.
- [RS91] Neil Robertson and Paul D Seymour. “Graph minors. X. Obstructions to tree-decomposition”. In: *Journal of Combinatorial Theory, Series B* 52.2 (1991), pp. 153–190.
- [RS95] Neil Robertson and Paul D Seymour. “Graph minors. XIII. The disjoint paths problem”. In: *Journal of combinatorial theory, Series B* 63.1 (1995), pp. 65–110.
- [Rus73] T Benny Rushing. *Topological embeddings*. Academic Press, 1973.
- [Sha+95] Farhad Shahrokhi, Ondrej Skora, László A Székely, and I Vrt’o. “Crossing numbers: bounds and applications”. In: *Intuitive geometry* 6 (1995), pp. 179–206.
- [Sti93] John Stillwell. *Classical topology and combinatorial group theory*. second. New York: Springer-Verlag, 1993.
- [Tho89] Carsten Thomassen. “The graph genus problem is NP-complete”. In: *Journal of Algorithms* 10.4 (1989), pp. 568–576.
- [Tho90] Carsten Thomassen. “Embeddings of graphs with no short noncontractible cycles”. In: *Journal of Combinatorial Theory, Series B* 48.2 (1990), pp. 155–177.
- [Tho92] Carsten Thomassen. “The Jordan-Schönflies theorem and the classification of surfaces”. In: *The American Mathematical Monthly* 99.2 (1992), pp. 116–130.
- [Veg04] Gert Vegter. “Computational topology”. In: *Handbook of Discrete and Computational Geometry*. Ed. by Jacob E. Goodman and Joseph O’Rourke. second. CRC Press LLC, 2004. Chap. 32, pp. 517–536.

- 
- [Web67] Claude Weber. “Plongements de polyedres dans le domaine metastable”. In: *Commentarii Mathematici Helvetici* 42.1 (1967), pp. 1–27.
- [You63] John William Theodore Youngs. “Minimal imbeddings and the genus of a graph”. In: *Journal of Mathematics and Mechanics* (1963), pp. 303–315.

