

Mesures de similarité pour cartes généralisées

Camille Combier

▶ **To cite this version:**

Camille Combier. Mesures de similarité pour cartes généralisées. Autre [cs.OH]. Université Claude Bernard - Lyon I, 2012. Français. NNT : 2012LYO10272 . tel-00995382

HAL Id: tel-00995382

https://theses.hal.science/tel-00995382

Submitted on 23 May 2014

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.

UNIVERSITÉ CLAUDE BERNARD LYON 1

N° attribué par la bibliothèque
272-2012







THÈSE

pour obtenir le grade de
DOCTEUR de U.C.B.L.
Spécialité : **Informatique**
préparée au **Laboratoire d'InfoRmatique en Images et Système d'information**
dans le cadre de l'École Doctorale **Math-Info**

présentée et soutenue publiquement
par

Camille Combier

le 28-11-2012

Titre:
Mesures de similarité pour cartes généralisées

Directrice de thèse: **Christine Solnon, Professeure, INSA Lyon, LIRIS**
Co-directeur de thèse: **Guillaume Damiand, Chargé de Recherches HDR, CNRS, LIRIS**

Jury

M. Christophe, Fiorio, Polytech'Montpellier, LIRMM,	Rapporteur
M. Pascal Lienhardt, Professeur, Univ. Poitier, XLIM-SIC,	Rapporteur
M. Luc Brun, Professeur, ENSICAEN, GREYC,	Examinateur
Mme. Raphaëlle Chaine, Maître de conférences, Univ. Lyon 1, LIRIS,	Examinatrice
Mme. Christine Solnon, Professeure, INSA Lyon, LIRIS,	Directrice de thèse
M. Guillaume Damiand, Chargé de Recherches HDR, CNRS, LIRIS,	Co-Directeur de thèse

Remerciements

Tout d'abord, je tiens à exprimer mes plus sincères remerciements à mes directeurs de thèse, Christine Solnon et Guillaume Damiand pour leur encadrement, leurs conseils et leur bonne humeur qui ont rendu ces trois dernières années aussi agréables qu'intéressantes.

Je tiens à remercier ensuite Christophe Fiorio et Pascal Lienhardt pour avoir accepté d'être mes rapporteurs et pour leurs précieuses suggestions et remarques.
Je voudrais également remercier Luc Brun et Raphaëlle Chaîne pour avoir accepté de participer au jury de soutenance en tant qu'examinateurs et pour leurs questions pertinentes.
Merci à tous les membres de l'équipe M2DisCo pour leurs conseils et remarques tout au long de ces trois années de thèse. Je veux aussi remercier tout ceux que j'ai pu croiser, que ce soit au coin-café, à une conférence ou au détour d'une journée des thèses et que je n'ai pas nommé explicitement.
Je tiens aussi à remercier tout particulièrement l'ensemble de mes co-bureaux. Stéphane, Romain, Alain et Loïc pour leur bonne humeur et les innombrables 'activités' de la pause midi auxquelles nous avons pu nous livrer pendant les deux premières années de ma thèse. Un grand merci à Aurelien, Jean-David, Jeremy et François pour m'avoir supporté pendant cette dernière année de thèse un peu mouvementée. Un merci tout particulier à Elsa qui a accepté de me supporter en plus de la journée certains soirs pour des séances de footing exutoires. Je voulais aussi remercier ma famille pour leur soutien pendant l'ensemble de mes années d'études.

Je tiens aussi à remercier profondément celle qui m'a motivé durant ces dix dernières années et encore plus pendant ces trois années de thèse de part sa présence et son infinie patience. Celle qui m'a motivé avec ses bons petits plats, sa bienveillance et qui a bien voulu sacrifier de son temps pour relire ce manuscrit. En bref celle qui partage autant mes bons que mes mauvais moments... Merci Maud.

Table des matières

1 Introduction **3**

2 Mesurer la distance entre deux graphes **7**
2.1 Définitions et notations . . . 7
2.2 Isomorphismes et sous-isomorphismes de graphes . . . 9
2.3 Plus grand sous-graphe commun et distance d'édition . . . 11
2.4 Plongements de graphes . . . 17
2.5 Discussion . . . 19

3 Cartes généralisées **21**
3.1 Définitions et notations . . . 21
3.2 Isomorphisme et sous-isomorphisme de nG-cartes . . . 24
3.3 Motifs fréquents . . . 26
3.4 Discussion . . . 28

4 Définition de distances entre cartes généralisées **29**
4.1 Sous-cartes induites et sous-cartes partielles . . . 29
4.2 Distance basée sur la plus grande sous-carte commune . . . 32
4.3 Distance d'édition entre nG-cartes . . . 38
4.4 Relation entre d_{mcs} et d_{ed} . . . 41
4.5 nG-cartes étiquetées et distance d'édition . . . 48
4.6 Discussion . . . 50

5 Algorithmes d'approximation de distances entre nG-cartes **51**
5.1 Appariements et sous-cartes . . . 51
5.2 Calcul d'une approximation de la plus grande sous-carte commune . . . 58
5.3 Heuristiques pour choisir le prochain couple de brins à apparier . . . 61
5.4 Évaluation expérimentale de ApproxD . . . 68
5.5 Discussion . . . 75

6 Évaluation expérimentale **77**
6.1 Base d'images . . . 78
6.1.1 Modélisation d'images par des 2G-cartes et des graphes . . . 78
6.1.2 Définition des étiquettes et des coûts . . . 80
6.1.3 Description du jeu de données . . . 84
6.1.4 Classification d'images . . . 87

6.1.5 Comparaison des appariements construits 90
6.2 Base de maillages 95
6.2.1 Définition des étiquettes et coûts 100
6.2.2 Description des jeux de données 102
6.2.3 Classification de maillages 104
6.2.4 Comparaison des appariements construits 106
6.3 Discussion 110

7 Conclusion **113**

Résumé

Une carte généralisée est un modèle topologique permettant de représenter implicitement un ensemble de cellules (sommets, arêtes, faces , volumes, . . .) ainsi que l'ensemble de leurs relations d'incidence et d'adjacence au moyen de brins et d'involutions. Les cartes généralisées sont notamment utilisées pour modéliser des images et objets 3D. Á ce jour il existe peu d'outils permettant l'analyse et la comparaison de cartes généralisées. Notre objectif est de définir un ensemble d'outils permettant la comparaison de cartes généralisées.

Nous définissons tout d'abord une mesure de similarité basée sur la taille de la partie commune entre deux cartes généralisées, appelée *plus grande sous-carte commune*. Nous définissons deux types de sous-cartes, *partielles* et *induites*, la sous-carte induite doit conserver toutes les involutions tandis que la sous-carte partielle autorise certaines involutions à ne pas être conservées. La sous-carte partielle autorise que les involutions ne soient pas toutes conservées en analogie au sous-graphe partiel pour lequel les arêtes peuvent ne pas être toutes présentes. Ensuite nous définissons un ensemble d'opérations de modification de brins et de coutures pour les cartes généralisées ainsi qu'une distance d'édition. La distance d'édition est égale au coût minimal engendré par toutes les successions d'opérations transformant une carte généralisée en une autre carte généralisée. Cette distance permet la prise en compte d'étiquettes, grâce à l'opération de substitution. Les étiquettes sont posées sur les brins et permettent d'ajouter de l'information aux cartes généralisées. Nous montrons ensuite, que pour certains coûts notre distance d'édition peut être calculée directement à partir de la plus grande sous-carte commune.

Le calcul de la distance d'édition est un problème NP-difficile. Nous proposons un algorithme glouton permettant de calculer en temps polynomial une approximation de notre distance d'édition de cartes. Nous proposons un ensemble d'heuristiques basées sur des descripteurs du voisinage des brins de la carte généralisée permettant de guider l'algorithme glouton, et nous évaluons ces heuristiques sur des jeux de test générés aléatoirement, pour lesquels nous connaissons une borne de la distance.

Nous proposons des pistes d'utilisation de nos mesures de similarités dans le domaine de l'analyse d'image et de maillages. Nous comparons notre distance d'édition de cartes généralisées avec la distance d'édition de graphes, souvent utilisée en reconnaissance de formes structurelles. Nous définissons également un ensemble d'heuristiques prenant en compte les étiquettes de cartes généralisées modélisant des images et des maillages. Nous mettons en évidence l'aspect qualitatif de notre appariement, permettant de mettre en correspondance des zones de l'image et des points du maillages.

Abstract

A generalized map is a topological model that allows to represent implicitly different types of cells (vertices, edges, volumes, ...) and their relationship by using a set of darts and some involutions. Generalized maps are used to model 3D meshes and images. Anyway there exists only few tools to compare theses generalized maps. Our main goal is to define some tools tolerant to error to compare them.

We define a similarity measure based on the size of the common part of two generalized maps, called *maximum common submap*. Then we define two types of submaps, *partial* and *induced*, the induced submap needs to preserve all the involutions whereas the partial one can allow some involutions to be removed. Then we define a set of operations to modify a generalized map into another and the associated edit distance. The edit distance is equal to the minimal cost of all the sequences of operations that modify a generalized map into the other. This edit distance can use labels to consider additional information, with the operation called 'substitution'. Labels are set on darts. We next show a relation between our edit distance and the distance based on the maximum common submap.

Computing theses distance are a NP-hard problem. We propose a greedy algorithm computing an approximation of it. We also propose a set of heuristics based on the description of the neighborhoob of the darts to help the greedy algorithm. We try theses heuristics on a set of generalized maps randomly generated where a lower bound of the distance is known. We also propose some applications of our similarity measures in the image analysis domain. We compare our edit distance on generalized maps with the edit distance on graphs. We also define a set of labels specific on images and 3D meshes. And we show that the matching computed by our algorithm construct a link between images's areas.

Chapitre 1

Introduction

Contexte

La manipulation d'objets graphiques est aujourd'hui un des enjeux majeurs de notre société. Les images et les vidéos étant de plus en plus présentes et nombreuses, nous notons une augmentation du nombre de travaux visant à les manipuler et les analyser. La représentation d'objets en 3 dimensions est aussi très utilisée, par exemple pour modéliser et prévisualiser un objet avant sa fabrication.
Les structures de données en informatique graphique sont à la fois le moyen de représenter des objets graphiques, mais aussi un moyen de les manipuler. Par exemple, pour représenter une image, nous pouvons la modéliser avec un ensemble de pixels ayant des relations de voisinage et pour modéliser un objet 3D, nous pouvons utiliser un ensemble de faces avec des relations de voisinage. Ces relations de voisinage sont décrites par la structure de données modélisant l'objet. Elles sont nécessaires pour de nombreux traitements des objets. Par exemple pour détecter des formes dans une image, nous regroupons les pixels voisins en régions ayant des couleurs similaires. Pour ce procédé appelé segmentation, nous avons besoin de savoir quels sont les pixels voisins. Pour cela, nous pouvons utiliser un graphe d'adjacence de régions. Ce graphe est composé d'un sommet par région, et d'une arête entre deux sommets si les deux régions correspondantes sont adjacentes. La Fig. 1.1(c) donne un exemple de graphe d'adjacence de régions correspondant à la segmentation Fig. 1.1(b) de l'image Fig. 1.1(a).

Dans le cas de la modélisation d'objets 3D une structure très utilisée pour représenter des faces et leur voisinage reste le simplexe. Un simplexe est un triangle de dimension quelconque : un triangle est un 2-simplexe, un tétraèdre est un 3-simplexe... Un maillage surfacique est représenté par un ensemble de 2-simplexes ainsi que la relation d'adjacence entre les triangles. Un maillage volumique sera représenté par un ensemble de 3-simplexes et leur relation d'adjacence.

Cependant, la plupart de ces structures possèdent des limitations. Par exemple le graphe d'adjacence est une modélisation incomplète d'une segmentation : il modélise les relations d'adjacence entre les régions de la segmentation, mais ne permet pas de modéliser les positions relatives de ces régions. Il est ainsi possible que deux segmentations différentes soient modélisées par le même graphe d'adjacence. Pour la modélisation 3D, un maillage utilisant des simplexes n'est pas incomplet mais est une structure régulière. Le simplexe étant un triangle de dimension quelconque, cette structure ne permet pas de représenter des faces autres que des triangles, et des volumes autres que des tétraèdres. La Fig. 1.1(e) est un exemple de maillage volumique modélisant un

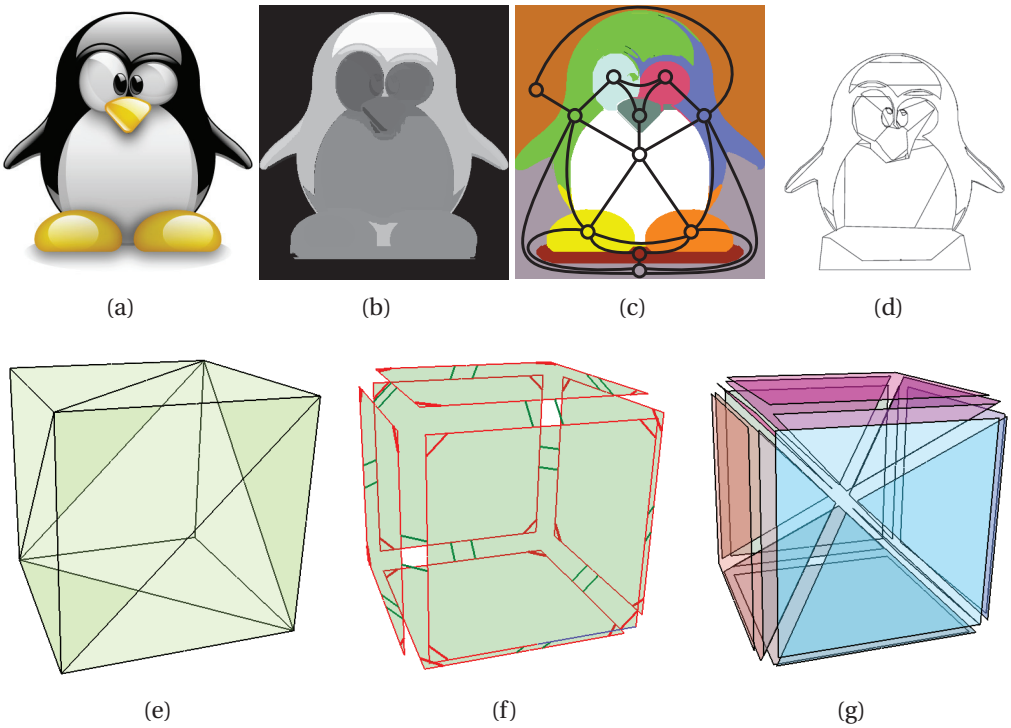


FIGURE 1.1 – *(a) Une image. (b) Une segmentation de l'image (a). (c) Exemple de modélisation de (b) avec un graphe d'adjacence de régions. (d) Exemple de modélisation de (b) avec une carte généralisée. (e) Exemple de cube modélisé avec des 2-simplexes. (f) Exemple de cube modélisé avec une carte généralisée. (g) Exemple d'un cube modélisé avec des pyramides.*

cube avec des 2-simplexes. Ne manipulant que des triangles, la structure ne permet pas des opérations telles que la fusion de faces, qui produit des faces non triangulaires. Nous souhaiterions pouvoir manipuler des maillages ayant des faces quelconques. En dimension supérieure, le problème est identique, les maillages volumiques sont représentés avec des 3-simplexes. Les cartes généralisées sont une structure de données répondant à ces limitations. En effet, elles représentent une subdivision de l'espace en cellules ainsi que la relation d'adjacence entre ces cellules. En 2D, les cellules modélisées sont les sommets, les arêtes et les faces.

Les cartes généralisées [Lie94] sont définies en toute dimension et peuvent donc représenter des cellules de dimensions supérieures à 2. Aussi, les cellules pouvant être modélisées ne sont pas obligatoirement régulières, il est donc possible en 2D de représenter des faces non triangulaires.

Un exemple de carte généralisée modélisant un cube est montré dans la Fig. 1.1(f) et un exemple de modélisation de maillage volumique d'un cube composé de 6 pyramides est illustré dans la Fig. 1.1(g).

En pratique, en 2D, les cartes généralisées peuvent être assimilées aux quad-edges, proposées dans [GS85]. Cette structure permet de représenter des maillages surfaciques et est aussi utilisée dans [HDMB07] pour modéliser des architectures.

Pour modifier les cartes généralisées, un ensemble d'opérations ont été définies : des opérations de triangulation, extrusion, opération booléennes, chanfreinage, produit cartésien, ... Mais aussi des opérations de suppression, de contraction, d'ajout et d'éclatement de cellules.

En informatique graphique, la mesure de similarité entre les objets est particulière-

ment utile cat elle permet d'effectuer de la classification, des recherches dans une base de données, ou de rechercher des points communs. Pour comparer des objets, nous comparons les structures qui les modélisent. Ainsi, pour mesurer la similarité entre deux images, nous pouvons mesurer la similarité entre deux graphes d'adjacence de régions, et pour mesurer la similarité entre deux maillages, nous pouvons mesurer la similarité des complexes simpliciaux. La pertinence de la classification ou de la recherche dépend donc très fortement d'une part des structures de données utilisées pour modéliser des objets, et d'autre part de la mesure de similarité utilisée pour comparer ces structures : il s'agit de disposer de mesures de similarité entre structures reflétant au mieux la similarité des objets modélisés par ces structures.

Il existe de nombreux travaux sur la comparaison et les mesures de similarité de graphes. En particulier, l'isomorphisme de graphes qui permet de décider de l'équivalence de deux graphes, tandis que l'isomorphisme de sous-graphe permet de décider de l'inclusion d'un graphe dans un autre. Ces méthodes ne sont cependant pas tolérantes aux erreurs, et c'est pourquoi d'autres méthodes plus flexibles ont été proposées. Une mesure de similarité entre graphes plus tolérante aux erreurs est basée sur le plus grand sous-graphe commun aux deux graphes. Le plus grand sous-graphe commun représente intuitivement la partie commune que nous pouvons retrouver dans les deux graphes comparés. Nous dénombrons de nombreuses déclinaisons de cette distance utilisant des variations du plus grand sous-graphe commun, ou des variations de la définition de la taille d'un graphe. Une autre mesure de similarité tolérante aux erreurs, et permettant de comparer des graphes étiquetés, est la distance d'édition, qui est basée sur le nombre d'opérations nécessaires pour modifier un graphe en un autre. Cette distance est une adaptation de la distance de Levenshtein sur les chaînes.

S'il existe de nombreux travaux sur la comparaison de graphes, il en existe beaucoup moins pour les cartes généralisées : pour les cartes généralisées, seule la définition de l'isomorphisme et du sous-isomorphisme existent, et une unique méthode de comparaison basée sur la recherche de motifs communs dans une base de cartes [GDS11]. C'est pourquoi nous nous proposons dans cette thèse de définir de nouvelles méthodes pour comparer les cartes généralisées.

Plan

L'objectif de cette thèse est donc de définir des mesures de similarités entre cartes généralisées. Pour cela nous nous sommes inspirés des travaux existants sur les graphes. Nous débuterons par un état de l'art des mesures de similarités utilisant les graphes dans le Chap. 2. Ensuite nous présenterons dans le Chap. 3 les cartes généralisées ainsi que l'ensemble des méthodes de comparaison existantes. A l'instar du plus grand sous-graphe commun nous définirons la plus grande sous-carte commune représentant la partie commune entre deux cartes généralisées dans le Chap. 4. Nous en déduirons une distance paramétrable permettant de choisir entre deux types de sous-cartes (induites ou partielles), et de pondérer l'importance des coutures par rapport aux brins. Nous montrerons que cette distance permet de définir un espace métrique sur l'ensemble des cartes généralisées. Ensuite nous proposerons un ensemble d'opérations de modification de cartes généralisées, et une distance d'édition basée sur l'ensemble d'opérations de coût minimum pour modifier une carte en une autre. Nous ajouterons la possibilité d'étiqueter des cartes, et adapterons la distance d'édition pour la prise en compte de ces étiquettes. Nous montrerons qu'il est possible de calculer la distance d'édition pour certains coûts en connaissant la plus grande sous-carte commune. En-

fin dans le Chap. 5 nous proposerons un algorithme construisant un appariement de brins pour lequel nous montrerons qu'il est possible d'extraire une sous-carte commune. Nous ajouterons un ensemble d'heuristiques pour orienter la construction de l'appariement afin d'obtenir un bon candidat à la plus grande sous-carte commune et les validerons sur un jeu de données généré aléatoirement. Nous proposerons et validerons dans le Chap. 6 plusieurs heuristiques adaptées à d'autres données telles que les images et les maillages, et montrerons que l'on peut utiliser les cartes pour effectuer de la classification et de la mise en correspondance de composantes.

Chapitre 2

Mesurer la distance entre deux graphes

Sommaire

2.1 Définitions et notations 7
2.2 Isomorphismes et sous-isomorphismes de graphes 9
2.3 Plus grand sous-graphe commun et distance d'édition 11
2.4 Plongements de graphes 17
2.5 Discussion 19

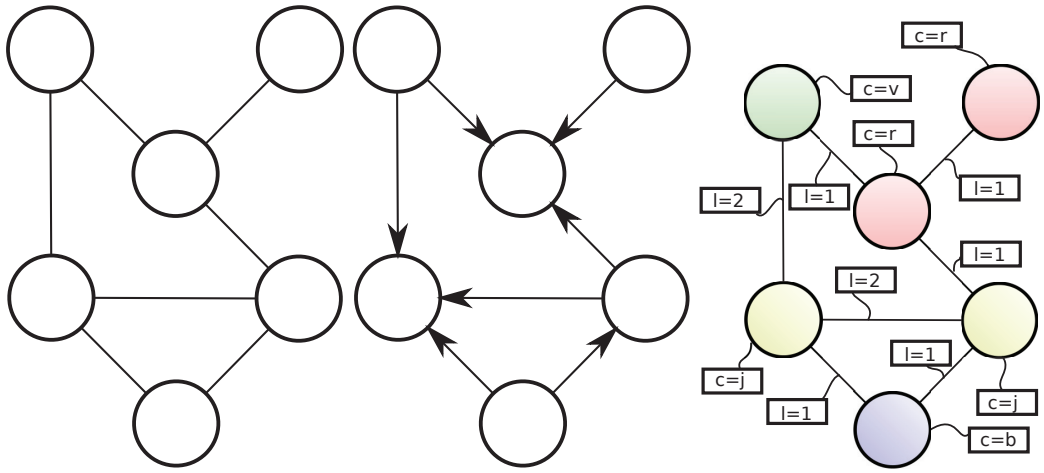
La pertinence d'une mesure de similarité entre objets dépend non seulement de la méthode de représentation mais aussi de la mesure de similarité considérée. Les graphes sont souvent utilisés pour modéliser des objets. Dans ce chapitre nous présentons tout d'abord différentes définitions et notations nécessaires autour des graphes puis nous présentons différentes méthodes de la littérature destinées à les comparer : tout d'abord des méthodes exactes, comme l'isomorphisme et le sous-isomorphisme, puis une méthode plus tolérante aux erreurs qui est la distance basée sur le plus grand sous-graphe commun, et enfin une méthode plus opérationnelle nommée distance d'édition. Ces travaux ont été des inspirations majeures pour cette thèse. Pour finir ce chapitre, nous présentons de nouvelles méthodes qui consistent à plonger les graphes dans des espaces vectoriels pour profiter des outils propres à ces espaces.

2.1 Définitions et notations

Un graphe est une structure de données représentant une relation binaire (arêtes) entre des sommets.

Définition 1 (Graphe)
Un graphe est un couple $G = (S, A)$ tel que
- *S est un ensemble fini de sommets ;*
- *$A \subseteq S \times S$ est une relation binaire sur S. Si la relation A est symétrique alors les éléments de A sont appelés des arêtes et le graphe est dit non-orienté ; sinon les éléments de A sont appelés des arcs et le graphe est dit orienté.*

Dans la suite de cet état de l'art, seul le cas des graphes non-orientés sera traité. Les notions peuvent être facilement étendues au cas des graphes orientés.



(a) Graphe non-orienté (b) Graphe Orienté (c) Graphe non orienté étiqueté

FIGURE 2.1 – *Quelques exemples de graphes.*

À chaque sommet ou arête il est possible d'associer un descripteur. Ce descripteur est appelé étiquette et permet d'ajouter des informations aux graphes.

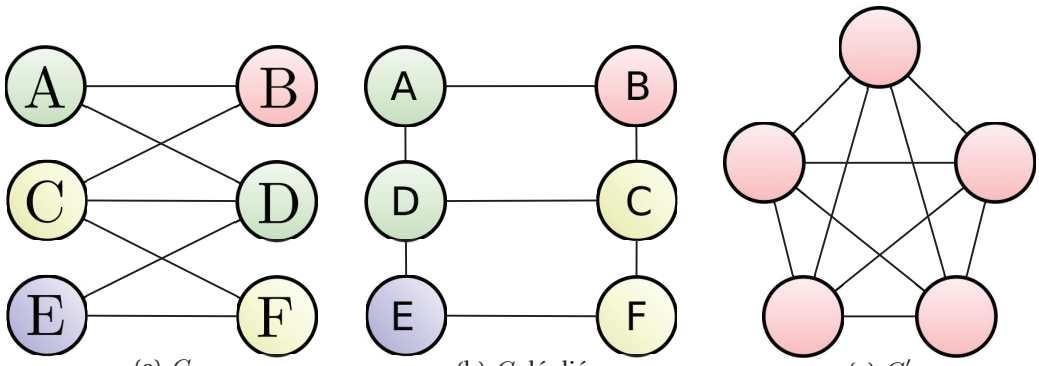
Définition 2 (Graphe étiqueté)
Un graphe étiqueté est un tuple $G = (S, A, E, e)$ *tel que*

- S *est un ensemble fini de sommets ;*
- $A \subseteq S \times S$ *est un ensemble d'arêtes ;*
- E *est l'ensemble des différentes étiquettes possibles ;*
- $e : S \cup A \rightarrow E$*, est une fonction qui associe une étiquette à chaque sommet et chaque arête.*

Étant donné un graphe G, nous notons $Sommets(G)$ l'ensemble des sommets de G et $Aretes(G)$ l'ensemble des arêtes de G.

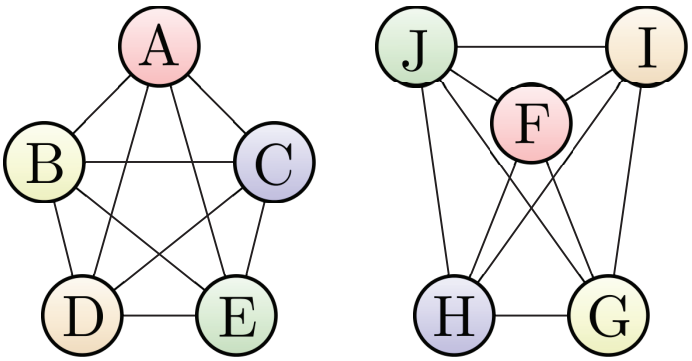
Dans la Fig. 2.1, nous avons représenté un graphe non-orienté (Fig. 2.1(a)), un graphe orienté (Fig. 2.1(b)) et un graphe possédant des étiquettes (couleurs et valeurs) (Fig. 2.1(c)).

La littérature différencie différents types de graphes, et en particulier les graphes planaires qui sont des graphes pour lesquels il existe un plongement dans le plan tel que leurs arêtes ne s'intersectent pas (en dehors des sommets). Un plongement de graphe dans un plan est équivalent au dessin de ce dernier sur un plan : cela revient à donner des coordonnées 2D à chacun des sommets. Ces graphes sont des graphes très particuliers pour lesquels les problèmes ne possèdent pas toujours la même complexité que pour les graphes non planaires. Notez que pour un graphe planaire, il peut exister plusieurs plongements tels que les arêtes ne s'intersectent pas, on nommera *graphe plan* un graphe plongé dans un plan et pour lequel les arêtes ne s'intersectent pas. Dans la Fig. 2.2, nous donnons des exemples de graphes planaires. Fig. 2.2(a) et Fig. 2.2(b) représentent 2 plongements différents du même graphe. Notez que le plongement de Fig. 2.2(b) ne possède pas de couples d'arêtes qui s'intersectent : c'est un graphe plan. Par contre, la Fig. 2.2(c) représente une clique (ensemble de sommets deux à deux adjacents) de taille 5, graphe non planaire.



(a) G (b) G déplié (c) G'

FIGURE 2.2 – *(a) et (b) représentent un même graphe planaire, avec deux plongements différents. Le plongement proposé par (b) nous prouve que le graphe est planaire et est un graphe plan. Pour (c) par contre, il n'existe pas de plongement sans intersection d'arêtes : le graphe n'est pas planaire.*



(a) G (b) G'

v	$f(v)$
A	F
B	G
C	H
D	I
E	J

(c) $Isomorphisme$

FIGURE 2.3 – *G et G' sont isomorphes et f est un isomorphisme de G vers G'.*

2.2 Isomorphismes et sous-isomorphismes de graphes

L'*isomorphisme de graphe* permet de décider de l'équivalence de graphes. C'est une fonction bijective des sommets du premier graphe vers les sommets du second telle que toutes les arêtes sont conservées.

Définition 3 (Isomorphisme de graphes)
Soient deux graphes $G = (S, A)$ et $G' = (S', A')$. Un isomorphisme entre G et G' est une fonction bijective $f : S \to S'$ telle que :
– $\forall (u, v) \in S \times S, (u, v) \in A \Leftrightarrow (f(u), f(v)) \in A'$.

Deux graphes sont dits isomorphes s'il existe un isomorphisme entre eux. Dans la Fig. 2.3 nous montrons deux graphes G et G' isomorphes : la fonction f est une fonction bijective entre leurs sommets telle que toutes les arêtes soient conservées.

Si le problème de la recherche d'un isomorphisme de graphe appartient clairement à la classe NP, il n'existe à ce jour ni algorithme polynomial pour le résoudre, ni preuve de NP-complétude. Une nouvelle classe de complexité a été définie pour ce problème : $Isomorphe - complet$. Néanmoins, le problème de l'isomorphisme dans le cas de graphes planaires a été montré comme faisant partie de la classe P, et un algorithme polynomial est proposé [HW74] pour calculer cet isomorphisme.

Dans le cas général, le calcul de l'isomorphisme de graphe, bien que $Isomorphe - complet$ est en pratique facilement résolu dans la grande majorité des cas par des algorithmes comme *Nauty* [McK08]. Avec *Nauty,* McKay propose une méthode de résolution dans le cas de graphes étiquetés en utilisant une représentation canonique des graphes. Cordella et al proposent [PCFSV04] un algorithme de calcul d'isomorphisme de graphe dédié aux graphes de grande taille en utilisant un appariement calculé à partir d'une représentation d'état.

Le second outil pour comparer des graphes est le *sous-isomorphisme de graphes,* ou *isomorphisme de sous-graphe* qui permet de décider de la relation d'inclusion entre graphes. Dans la littérature, on distingue deux types de sous-isomorphisme de graphes. Le sous-isomorphisme engendré par les sommets (sous-graphe induit) et le sous-isomorphisme engendré par les arêtes (sous-graphe partiel). Plus précisément, un graphe $G' = (S', A')$ est un sous-graphe induit du graphe $G = (S, A)$ s'il est composé d'un sous-ensemble $S' \subseteq S$ des sommets de G et que toutes ses arêtes incidentes à deux sommets de S' sont conservées. Dans le cas partiel, certaines arêtes de G peuvent ne pas être conservées dans G.

Définition 4 (Sous-graphe induit)
Soient deux graphes $G = (S, A)$ *et* $G' = (S', A')$. G' *est un* sous-graphe induit *de* G *si :*
- $S' \subseteq S$;
- $A' = A \cap S' \times S'$.

Définition 5 (Sous-graphe partiel)
Soient deux graphes $G = (S, A)$ *et* $G' = (S', A')$. G' *est un* sous graphe partiel *de* G *si :*
- $S' \subseteq S$;
- $A' \subseteq A \cap S' \times S'$.

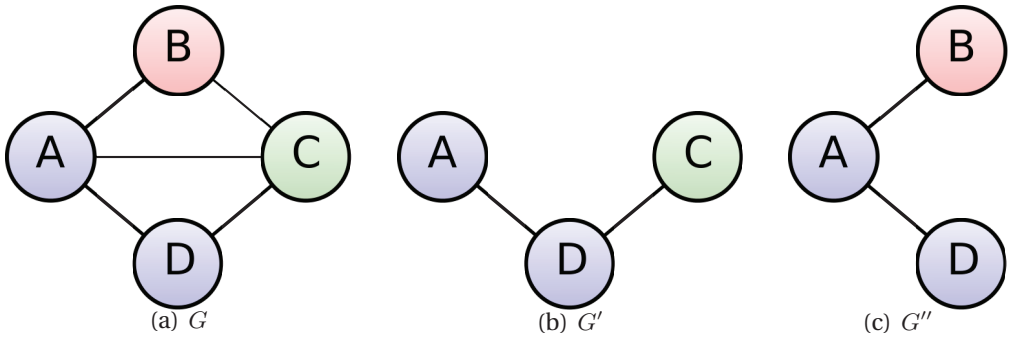
Dans la Fig. 2.4 nous donnons deux exemples de sous-graphes. De ces définitions de sous-graphes, nous en déduisons l'*isomorphisme de sous-graphe* permettant de décider de la relation d'inclusion entre deux graphes distincts.

Définition 6 (Isomorphisme de sous-graphe)
Soient deux graphes $G = (S, A)$ *et* $G' = (S', A')$. *Il existe un* isomorphisme de sous-graphe *induit (resp. partiel) de* G' *vers* G *s'il existe un sous-graphe induit (resp. partiel) de* G *isomorphe à* G'.

L'existence d'un isomorphisme de sous-graphe induit se traduit par l'existence d'une bijection f entre un sous-ensemble S_1 des sommets de G et les sommets de G' telle que toutes les arêtes de G entre deux sommets de S_1 soient conservées entre leurs images par f.

De manière équivalente, il existe un isomorphisme de sous-graphe partiel s'il existe une bijection f entre un sous-ensemble S_1 des sommets de G et les sommets de G' telle qu'il n'existe pas une arête entre deux sommets de S' qui n'existait pas entre deux sommets de S par la bijection f^{-1}.

Dans le cas général, la recherche d'un isomorphisme de sous-graphes (partiel ou induit) est un problème NP-complet. Le problème de recherche d'un isomorphisme de sous-arbres a également été montré comme faisant partie de la classe des problèmes NP-complet par Garey et Johnson dans [GJ79]. Les arbres étant des graphes planaires, le problème de recherche d'isomorphisme de sous-graphe dans le cas de graphes planaires est aussi NP-complet. Cependant, il existe certaines classes de graphes pour lesquels le problème peut être résolu par un algorithme polynomial. C'est



(a) G (b) G' (c) G''

FIGURE 2.4 – *G'' est un sous-graphe induit de G car pour les sommets A, B et D les arêtes sont conservées. G' n'est pas un sous-graphe induit de G car l'arête entre A et C n'est pas conservée, par contre c'est un sous-graphe partiel de G car toutes les arêtes de G' existent dans G.*

le cas des graphes planaires 3-connexes [Epp95] ou encore des graphes planaires extérieurs 2-connexes (tous les sommets appartiennent à la face extérieur du tracé) [Sys82]. [Ull76] propose un des premiers algorithmes dédiés au problème de décision du sous-isomorphisme en utilisant des arbres de recherche. [CFSV04] propose un algorithme dédié aux graphes de grande taille. Enfin, [Sol10] propose une méthode utilisant la programmation par contrainte.

La recherche d'isomorphismes ou de sous-isomorphismes de graphes peut être simplement étendue au cas des graphes étiquetés, pour cela nous considérons que deux sommets sont équivalents si leurs étiquettes sont égales. Dans le cas d'étiquettes numériques nous pouvons aussi par exemple relâcher la contrainte sur les étiquettes en autorisant des valeurs d'étiquettes proches plutôt que leur égalité.

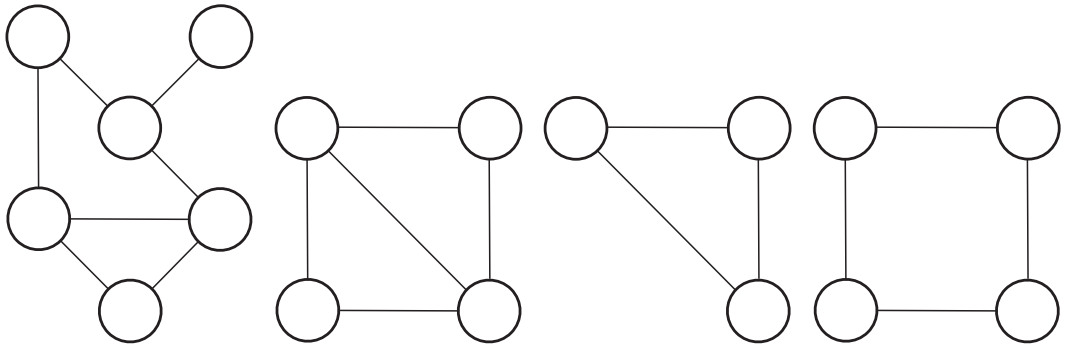
2.3 Plus grand sous-graphe commun et distance d'édition

La notion d'inclusion entre graphes peut être utilisée pour rechercher la partie commune (donc sous-graphe) de plusieurs graphes. Pour cela, nous pouvons définir la notion de *sous-graphe commun*. Un graphe G est un sous-graphe (induit ou partiel) commun à deux graphes G_1 et G_2 s'il existe des sous-graphes (induits ou partiels) de G_1 et G_2 qui sont isomorphes à G.

Définition 7 (Sous-graphe commun (induit ou partiel))
Soient deux graphes $G = (S, A)$ et $G' = (S', A')$. G'' est un sous-graphe commun induit (resp. partiel) de G et G' si G'' est isomorphe à un sous-graphe induit (resp. partiel) de G et à un sous-graphe induit (resp. partiel) de G'.

Le sous-graphe commun correspond à une partie commune aux deux graphes. Dans [BS98], Bunke et Shearer proposent une distance basée sur la taille du plus grand sous-graphe induit commun de deux graphes. L'intuition est que la partie commune de deux graphes est d'autant plus grande que les graphes sont proches. La taille utilisée est basée uniquement sur le nombre de sommets. Nous pouvons aussi utiliser le nombre d'arêtes ou toute combinaison linéaire des deux en fonction du type de sous-graphe considéré. Nous notons $|G|$ la taille d'un graphe G.

En conséquence, nous pouvons définir le plus grand sous-graphe induit (resp. partiel) commun, comme le sous-graphe induit (resp. partiel) commun ayant la plus grande taille.



(a) G (b) G' (c) $mcs^i(G,G')$ (d) $mcs^p(G,G')$

FIGURE 2.5 – *En définissant* $|G| = |Sommets(G)|$, $mcs^i(G,G')$ *est le plus grand sous-graphe commun induit de* G *et* G' *et* $d(G,G') = \frac{3}{6} = 0.5$. *En définissant* $|G| = |Aretes(G)|$, $mcs^p(G,G')$ *est le plus grand sous-graphe commun partiel de* G *et* G' *et* $d(G,G') = \frac{4}{7} = 0.43$.

Définition 8 (Plus grand sous-graphe commun)
Soient trois graphes, G, G_1 *et* G_2. G *est le plus grand sous-graphe commun de* G_1 *et* G_2 *si* G *est un sous-graphe commun de* G_1 *et de* G_2 *et s'il n'existe pas* G' *sous-graphe commun de* G_1 *et* G_2 *tel que* $|G'| > |G|$.
Nous notons le plus grand sous-graphe commun de G_1 *et* G_2 *:* $mcs(G_1,G_2)$.

La distance entre deux graphes utilisant le plus grand sous-graphe commun est définie dans [BS98] de la façon suivante :

Définition 9 (Distance entre graphes basée sur un plus grand sous-graphe commun)
La distance entre deux graphes non-vides G_1 *et* G_2 *est définie par*
$d(G_1,G_2) = 1 - \frac{|mcs(G_1,G_2)|}{max(|G_1|,|G_2|)}$

Bunke et Shearer ont montré que cette distance, quand la taille considérée est le nombre de sommets des graphes et que le sous-graphe considéré est induit, permet de définir un espace métrique sur l'ensemble des graphes.

Définition 10 (Espace métrique)
L'espace $(\mathcal{E},d)$ *est un espace métrique si* $\mathcal{E}$ *est un ensemble non vide et* d *une distance sur* $\mathcal{E}$. *Nous appelons* distance *sur un ensemble* $\mathcal{E}$, *une application* d *de* $\mathcal{E}^2$ *dans* $\mathbb{R}^+$ *telle que, pour tout* x,y,z *de* $\mathcal{E}$, d *soit :*

- *(i) Positive :* $d(x,y) \geq 0$;
- *(ii) Symétrique :* $d(x,y) = d(y,x)$;
- *(iii) Séparative :* $d(x,y) = 0$ *ssi* x *et* y *sont indicernables ;*
- *(iv) Inégalité triangulaire :* $d(x,y) + d(y,z) \geq d(x,z)$.

La Fig. 2.5 illustre deux graphes G et G', leurs plus grand sous-graphes communs induit et partiel, et donne la distance correspondante.
D'autres métriques ont été proposées, dont une basée sur le plus petit super-graphe commun [FV01], c'est-à-dire le plus petit graphe pour lequel les deux graphes comparés sont des sous-graphes. [FV01] montre un lien direct entre le plus petit super-graphe et le plus grand sous-graphe commun.

Le calcul de cette distance repose sur la capacité à calculer le plus grand sous-graphe commun de deux graphes. Ce problème, plus général que le problème d'isomorphisme de sous-graphe, est NP-Difficile. Pour calculer le plus grand sous-graphe

commun, McGregor [McG82] propose un algorithme basé sur un arbre de recherche dont chaque noeud est un sous-graphe commun des deux graphes d'entrée. Chaque niveau de l'arbre représente l'ajout d'un couple de sommets à l'appariement. Un algorithme en profondeur est utilisé pour trouver le plus grand sous-graphe commun se situant sur la feuille ayant le plus d'ancètres. L'algorithme de Durand-Pasari [DP99] propose de ramener le problème de recherche de plus grand sous-graphe commun au problème de recherche de clique maximum. Pour cela il crée un graphe produit et utilise la recherche de clique maximum dans ce nouveau graphe. Enfin, Balas Yu [BY86] propose de rechercher la plus grande clique maximum en utilisant une méthode de coloriage.

Le plus grand sous-graphe partiel est aussi utilisé pour mesurer la similarité entre graphes. C'est le cas de [RGW02] qui calcule un candidat pour le plus grand sous-graphe commun partiel en utilisant une taille basée sur une combinaison linéaire du nombre d'arêtes et de sommets. En contrepartie, cette distance ne vérifie pas forcément les propriété de métrique. Enfin, dans [NS11] Ndiaye et Solnon proposent d'utiliser la programmation par contraintes pour rechercher le plus grand sous-graphe induit ou partiel. Cette nouvelle méthode considère l'ensemble des sous-graphes comme l'espace de recherche, et par un élagage ils recherchent le plus grand sous-graphe commun.

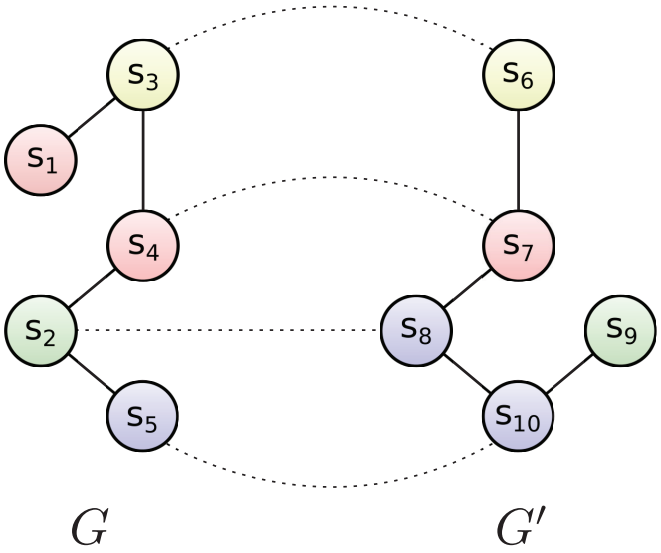
Distance d'édition

Toujours dans l'objectif de calculer la distance entre deux graphes, on peut se placer d'un point de vue plus opérationnel. Dans [Bun97] H. Bunke définit une nouvelle mesure de similarité entre graphes appelée distance d'édition. C'est une distance inspirée de la distance d'édition de chaînes de caractères ou distance de Levenshtein [Lev65]. La distance d'édition entre G et G' est définie comme la suite d'opérations la moins coûteuse pour rendre le graphe G isomorphe au graphe G'. Les opérations considérées sont l'insertion, la suppression et la substitution de sommets et d'arêtes. Cette distance permet la prise en compte d'étiquettes, qui permettent d'ajouter de l'information à la structure du graphe. Pour cela, l'opération de substitution permet de modifier une étiquette en une autre.

Pour définir la distance d'édition nous avons besoin de la notion d'appariement, permettant de mettre en relation les sommets de deux graphes distincts. Nous distinguons les appariements univoques qui mettent en relation chaque sommet d'un graphe avec au plus un sommet de l'autre graphe, et les appariements multivoques qui permettent de mettre en relation un sommet avec plusieurs sommets de l'autre graphe.

Définition 11 (Appariement de graphes)
Soient deux graphes $G = (S, A)$ et $G' = (S', A')$. La relation R sur $S \times S'$ est un appariement entre les graphes G et G'.
Il est dit univoque si $\forall (s_1, s'_1), (s_2, s'_2) \in R, s_1 = s_2 \Leftrightarrow s'_1 = s'_2$, il est dit multivoque sinon.

Étant donné un appariement univoque m entre deux graphes G et G', et un sommet s de G (resp. de G') nous notons $m(s)$ le sommet de G' (resp. G) apparié à s par m. Étant donné deux graphes $G = (S, A)$ et $G' = (S', A')$, un appariement à tolérance d'erreurs (noté $ecgm$ dans la litterature) est un appariement univoque $m \subseteq S \times S'$. De cet appariement est déduit une suite d'opérations entre les deux graphes G et G' permettant d'obtenir un graphe G" à partir du graphe G tel que G" est isomorphe à G'. Dans la suite, nous nommons $\hat{S}$ (resp. $\hat{S'}$) le sous-ensemble des sommets de S (resp. S') apparaissant dans l'appariement m. De manière identique, nous nommons $\hat{A}$ (resp. $\hat{A'}$) le



$$m = \{(s_3, s_6), (s_4, s_7), (s_2, s_8), (s_5, s_{10})\}$$

FIGURE 2.6 – *Deux graphes G et G', et un appariement m. Le sommet s_1 et l'arête (s_1, s_3) sont supprimés, le sommet s_9 et l'arête (s_9, s_{10}) sont insérés, les sommets s_2, s_3, s_4 et s_5 sont respectivement substitués aux sommets s_8, s_6, s_7,s_9 et s_{10}. Les arêtes $(s_3, s_4), (s_4, s_2)et(s_2, s_5)$ sont respectivement substituées par les arêtes $(s_6, s_7), (s_7, s_8)$ et (s_8, s_{10}).*

sous-ensemble des arêtes de A (resp. A') dont les deux sommets s_1 et s_2 (resp. s'_1 et s'_2) appartiennent à $\hat{S}$ (resp. $\hat{S}'$) de sorte qu'il existe une arête dans A' (resp. A) entre $m(s_1)$ et $m(s_2)$ (resp. $m(s'_1)$ et $m(s'_2)$), i.e., $\hat{A} = \{(s_1, s_2) \in \hat{S} \times \hat{S} \cap A | (m(s_1), m(s_2)) \in A'\}$ et de manière identique $\hat{A}' = \{(s'_1, s'_2) \in \hat{S}' \times \hat{S}' \cap A' | (m(s'_1), m(s'_2)) \in A\}$.

Définition 12 (Appariement de graphes à tolérance d'erreur)
Soient deux graphes étiquetés $G = (S, A, E, e)$ et $G' = (S', A', E', e')$ et m un appariement univoque entre G et G'. L'ensemble des opérations d'édition associées à m est :

- $\forall s \in \hat{S}, s$ *est substitué à* $m(s)$ *;*
- $\forall s \in S \setminus \hat{S}, s$ *est supprimé ;*
- $\forall s' \in S' \setminus \hat{S}', s'$ *est inséré ;*
- $\forall (s_1, s_2) \in \hat{A}$, *alors* (s_1, s_2) *est substituée à l'arête* $(m(s_1), m(s_2))$.
- $\forall (s_1, s_2) \in A \setminus \hat{A}, (s_1, s_2)$ *est supprimée ;*
- $\forall (s'_1, s'_2) \in A' \setminus \hat{A}', (s'_1, s'_2)$ *est insérée.*

La Fig. 2.6 donne un exemple d'appariement de graphes à tolérance d'erreurs. L'application au graphe G des opérations d'édition associées à cet appariement nous permet d'obtenir un graphe isomorphe au graphe G'.

La substitution est une opération permettant de caractériser la distance entre les sommets (resp. arêtes). En effet, dans le cas d'un graphe étiqueté cette opération va permettre de définir un coût de transformation d'un sommet (resp. arête) avec une étiquette en un sommet (resp. arête) ayant potentiellement une étiquette différente.

La distance d'édition est définie à partir de cet appariement à tolérance d'erreurs, et

de fonctions de coûts. Nous définissons le coût c d'un appariement m :

$$c(m) = \sum_{s\in\hat{S}} c_{ss}(s) + \sum_{s\in S-\hat{S}} c_{sd}(s) + \sum_{s\in S'-\hat{S}'} c_{si}(s) + \sum_{(s_1,s_2)\in\hat{A}} c_{as}((s_1,s_2)) + \sum_{(s_1,s_2)\in A-\hat{A}} c_{ad}((s_1,s_2)) + \sum_{(s'_1,s'_2)\in A'-\hat{A}'} c_{ai}((s'_1,s'_2))$$

avec

- $c_{ss}(s)$ le coût de substitution d'un sommet $s \in \hat{S}$ en $m(s) \in \hat{S}'$
- $c_{si}(s)$ le coût d'insertion d'un sommet $s \in S' - \hat{S}'$
- $c_{sd}(s)$ le coût de suppression d'un sommet $s \in S - \hat{S}$
- $c_{as}((s_1,s_2))$ le coût de substitution d'une arête $(s_1,s_2) \in \hat{A}$ en $(m(s_1),m(s_2)) \in \hat{A}'$
- $c_{ai}((s_1,s_2))$ le coût d'insertion d'une arête $(s_1,s_2) \in A' - \hat{A}'$
- $c_{ad}((s_1,s_2))$ le coût de suppression d'une arête $(s_1,s_2) \in A - \hat{A}$

Nous pouvons ainsi en déduire la distance d'édition qui est le coût de l'appariement à tolérance d'erreur le moins coûteux entre les deux graphes.

Définition 13 (Distance d'édition)
La distance d'édition entre deux graphes G et G' est égale au coût minimum engendré par tous les appariements à tolérence d'erreurs de G à G'. $d_{edit}(G,G') = min\{c(m)|m$ est un appariement de graphe à tolérence d'erreur de G vers $G'\}$

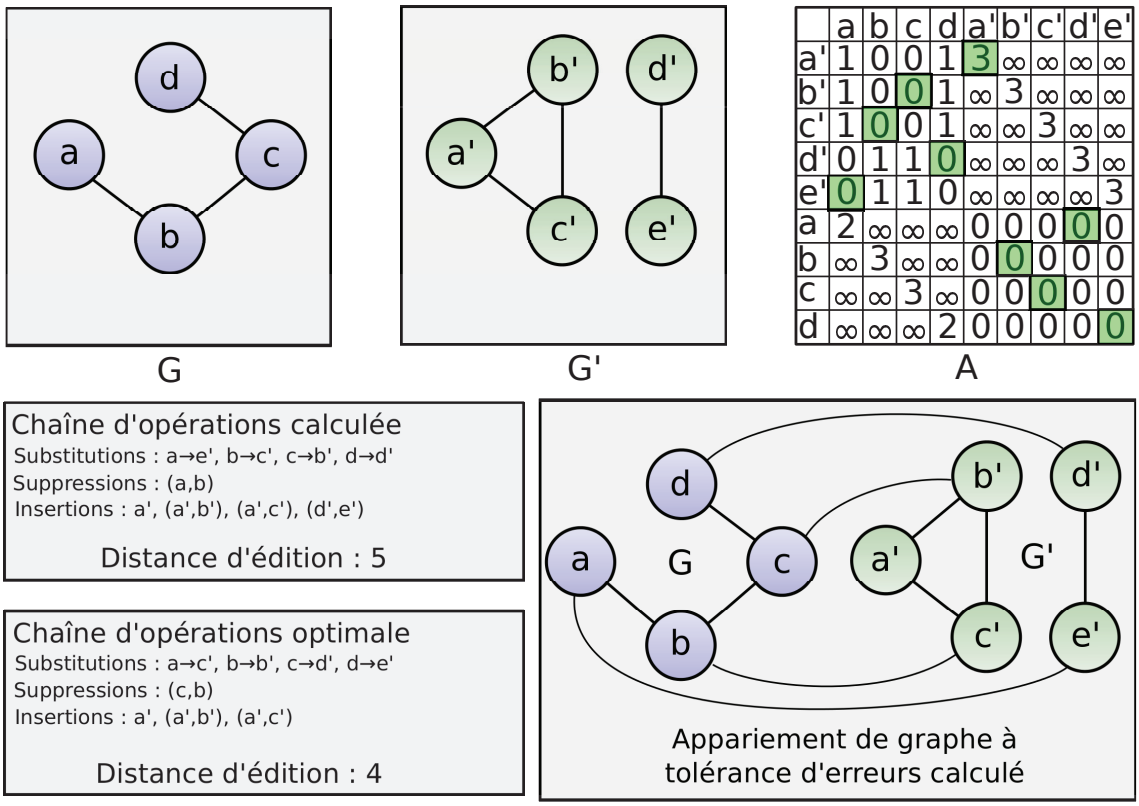
Dans [Bun97] H. Bunke met en évidence un lien entre le plus grand sous-graphe commun induit et la distance d'édition. En effet, il montre que si nous considérons une fonction de coût associant 1 à chaque sommet ou arête ajouté ou supprimé et 0 pour chaque substitution, alors étant donnés deux graphes $G = (S,A)$ et $G' = (S',A')$ la distance d'édition est égale à
$d_{edit}(G,G') = |Sommets(G)| + |Sommets(G')| - 2|Sommets(mcs(G,G'))|$. Ce résultat est primordial car il montre que dans ce cas le calcul de la distance d'édition se ramène au calcul du plus grand sous-graphe commun.

Calculer la distance d'édition est donc un problème NP-Difficile et de nombreux travaux ont cherché à approcher le plus grand sous-graphe commun pour pouvoir approximer la distance d'édition.

Pour cela, certains auteurs proposent des méthodes utilisant des algorithmes dits gloutons. Dans [CS03] les auteurs proposent un algorithme glouton qui, partant d'un appariement vide et guidé par une heuristique, ajoute successivement à cet appariement de nouveaux couples de sommets, jusqu'à ce qu'il n'y ait plus de couples de sommets améliorants. Ils déduisent de cet appariement une mesure de similarité entre les deux graphes.

Les méthodes à base de recherche locale permettent de remettre en cause une partie des choix réalisés, contrairement à l'algorithme glouton, et permettent souvent d'améliorer le résultat. Ainsi, [SS05] propose une recherche locale utilisant une liste taboue afin d'empêcher la recherche locale de boucler en reconsidérant les derniers choix réalisés. D'autres travaux proposent des méthodes à base de colonies de fourmis [SSSG06, SSG05]. Les colonies de fourmis utilisent une succession d'algorithmes gloutons qui utilisent des marqueurs (phéromones) pour mémoriser la qualité de la solution courante. Il est ainsi possible de jouer avec les probabilités pour améliorer une solution existante ou au contraire se diversifier en cherchant une autre solution.

Une méthode utilisant la programmation linéaire, dans le cas de graphes non-etiquetés est aussi proposée dans [JH06]. Les auteurs définissent une formulation du



	a	b	c	d	a'	b'	c'	d'	e'
a'	1	0	0	1	3	∞	∞	∞	∞
b'	1	0	0	1	∞	3	∞	∞	∞
c'	1	0	0	1	∞	∞	3	∞	∞
d'	0	1	1	0	∞	∞	∞	3	∞
e'	0	1	1	0	∞	∞	∞	∞	3
a	2	∞	∞	∞	0	0	0	0	0
b	∞	3	∞	∞	0	0	0	0	0
c	∞	∞	3	∞	0	0	0	0	0
d	∞	∞	∞	2	0	0	0	0	0

FIGURE 2.7 – *Pour comparer les deux graphes G et G', [RB09] construit la matrice A composée de 3 parties. Pour tout sommet i de G et tout sommet j' de G', $A_{ij'}$ donne le coût de substitution de i en j' et $A_{j'i} = 0$. Pour tout sommet i de G, A_{ii} donne le coût de suppression de i et pour tout sommet j de G tel que $i \neq j$, $A_{ij} = +\infty$. Pour tout sommet i' de G', $A_{i'i'}$ donne le coût d'insertion de i' et pour tout sommet j' de G' tel que $i' \neq j'$, $A_{i'j'} = +\infty$. Dans la partie supérieure gauche de A nous représentons ainsi les coûts de substitution, dans la partie supérieure droite les coûts d'insertion et dans la partie inférieure gauche les coûts de suppression. Le couplage de A de coût minimal est donné en vert. À partir de ce couplage, nous en déduisons une suite d'opérations pour modifier G en G'. Cette chaîne est composée de 1 suppression d'arête et de 4 insertions (1 sommet et 3 arêtes), or la chaîne minimale est composée de 1 suppression d'arête et 3 insertions (1 sommet et 2 arêtes).*

problème comme un programme linéaire binaire et donne une approximation de la distance d'édition en temps polynomial.

Enfin, plus récemment Riesen et Bunke [RB09] ont proposé un algorithme pour calculer rapidement une approximation de la distance d'édition en reformulant le problème en un problème d'affectation. Pour cela ils construisent une matrice composée de 3 parties. La partie en haut à gauche permet de décrire les coûts de substitutions, en bas à gauche les coûts de suppressions et en haut à droite les coûts d'insertions. En calculant une affectation (correspondant à un couplage minimal) de cette matrice avec l'algorithme de Munkres [Mun57], nous pouvons en déduire une suite d'opérations approximant la distance d'édition. La complexité de cet algorithme est $O(n^3)$ où n est le plus grand nombre de noeuds des graphes comparés. Cet algorithme ne génère néanmoins qu'une approximation de la distance d'édition du fait qu'il ne considère les choix que localement, et n'a pas de vue globale du problème posé. Un exemple d'exécution est présenté dans la Fig. 2.7.

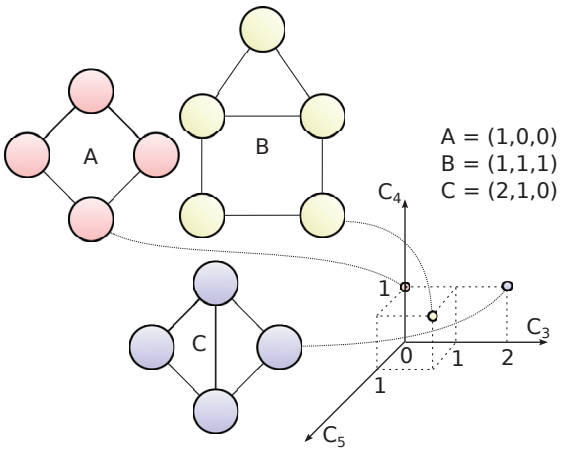


FIGURE 2.8 – *Illustration de plongement de graphes dans un espace 3D, sur les trois axes. Nous représentons le nombre de cycles élémentaires de taille* $3, 4$ *et* 5 *des graphes. Le vecteur pour chaque graphe est donc* $G = (C_3, C_4, C_5)$ *ou* C_i *est le nombre de cycles élémentaires de taille* i*.*

2.4 Plongements de graphes

Les distances présentées dans la Section 2.3 reposent sur le calcul du plus grand sous-graphe commun, qui est un problème NP-Difficile. Les plongements de graphes sont une alternative permettant d'éviter le calcul du plus grand sous-graphe ou de la distance d'édition.

Les plongements peuvent être utilisés pour mesurer la similarité entre graphes. L'idée principale est de représenter chaque graphe comme un point dans un espace vectoriel. Nous pouvons alors utiliser des mesures définies sur les espaces vectoriels.

Nous distinguons deux types de plongements, les plongements explicites qui décrivent explicitement la fonction de plongement des objets dans un espace vectoriel, et les plongements implicites qui ne nécessitent pas de calculer cette fonction.

Plongements de graphes explicites

Dans le cas des plongements explicites, nous définissons une fonction de projection des graphes dans un espace vectoriel.

Définition 14 (Plongement de graphes)
Un plongement de graphes est une fonction $\Phi : \mathcal{G} \to \mathbb{R}^n$ *d'un espace de graphes* $\mathcal{G}$ *vers un espace vectoriel de dimension* $n > 0$*.*

Chacune des dimensions de l'espace vectoriel représente une caractéristique du graphe, l'objectif étant d'utiliser les caractéristiques les plus appropriées. En voici quelques exemples :

- Nombre de sommets/arêtes ;
- Distribution des degrés des sommets ;
- Nombre de cycles élémentaires de longueur k ;
- Nombre de chemins de longueur k [HAA02] ;
- Valeurs propres issues des vecteurs propres de la matrice d'adjacence [LWHH03] ;
- ...

La Fig. 2.8 illustre le plongement de trois graphes dans un espace vectoriel. Dans cet exemple, nous plongeons les graphes dans un espace à 3 dimensions, chaque dimension représentant le nombre de cycles élémentaires d'une longueur donnée (3,4 et 5).

Pour calculer le plongement de chaque graphe, nous comptons le nombre de cycles élémentaires de longueurs 3,4 et 5 et nous en déduisons directement le point représentant le graphe dans l'espace vectoriel. Cette méthode permet de bénéficier des nombreux algorithmes de classification définis sur les espaces vectoriels [CHV99, BGV92].

De la même manière que pour les cycles, nous pouvons compter différents motifs au sein du graphe, [GBV11] propose par exemple de compter des 'treelets' qui sont des arbres, sous-graphes des graphes comparés. Cela s'applique principalement au domaine de la chémoinformatique (qui est le domaine liant la chimie et l'informatique) car ces 'treelets' sont une forme très présente dans les graphes représentant les molécules. [SVP+09, MP08] proposent de dénombrer les 'graphlets' qui sont des sous-graphes et qui sont moins spécifiques que les 'treelets'.

Au delà de l'utilisation de caractéristiques particulières aux graphes (cycles, chemins, ...) certaines méthodes sont basées sur les méthodes spectrales [Chu97, LWH03, RKH07]. Par exemple [LWHH03] propose de transformer la matrice d'adjacence en vecteurs propres, et de considérer les propriétés spectrales de cette dernière au lieu des propriétés du graphe.

Plongements de graphes implicites

Les plongements de graphes explicites nécessitent de calculer la projection des graphes dans l'espace vectoriel. En opposition les plongements de graphes implicites permettent de calculer la similarité entre deux graphes sans être obligé de calculer ce plongement.

Le plongement d'un graphe implicite (ou fonction de noyau) est une fonction $K(G, G')$ retournant la valeur du produit scalaire des plongements de G et G' dans un espace vectoriel sans calculer ces plongements : c'est le "kernel trick". Les contraintes sur cette fonction kernel sont qu'elle doit être continue, symétrique et semi définie positive. Le Theorème de Mercer [Mer09] montre que nous pouvons toujours trouver un espace vectoriel de grande dimension tel que cette fonction s'exprime comme un produit scalaire dans cet espace. Dans [KTI03], Kashima et al proposent de caractériser un graphe en comptant le nombre de chemins dans les graphes comparés. Pour cela, ils utilisent le fait que le nombre de chemins de longueur k peut être calculé en élevant la matrice d'adjacence à la puissance k. Ils utilisent cette technique pour compter le nombre de chemins dans le graphe $G \times G'$ (graphe produit de G et G') calculant ainsi le produit scalaire des plongements sans calculer explicitement tous les chemins communs. Ils montrent que chaque chemin dans $G \times G'$ correspond à un chemin dans G et G'.

La Fig. 2.9 illustre le calcul du nombre de chemins de taille k en commun entre deux graphes en calculant le graphe produit, puis en élevant la matrice d'adjacence de ce graphe à la puissance k. Dès lors que nous disposons d'une fonction noyau, nous pouvons utiliser les mesures et outils définis sur les espaces vectoriels qui utilisent uniquement le produit scalaire comme, par exemple, l'algorithme de classification SVM [Vap98].

Ces méthodes sont particulièrement utilisées pour classifier les objets. Nous notons les travaux de [GBV11] permettant la classification de molécules en utilisant ces plongements implicites . Il existe plusieurs noyaux, dont le noyau de diffusion [KL02, KStC03], le noyau de convolution [Hau99, Wat99] et les noyaux de chemins [BOS+05, GFW03]. Enfin, une synthèse de l'utilisation des noyaux de graphes pour comparer les objets a été publiée par Bunke H. et Riesen K. dans [BR11]. Nous notons que l'utilisation de

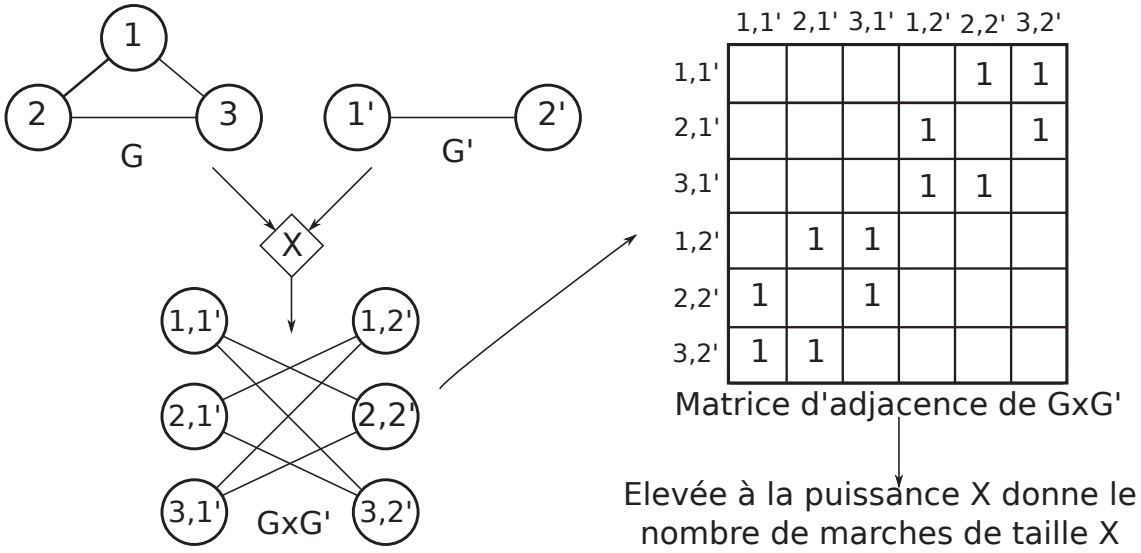


FIGURE 2.9 – *Illustration des plongements de graphes implicites, avec le produit vectoriel de deux graphes, et sa représentation en matrice d'adjacence.*

noyaux, bien que récente, est maintenant de plus en plus fréquente, et que les résultats qui en découlent sont particulèrement bons.

2.5 Discussion

Dans ce chapitre nous avons présenté les notions de base autour des graphes, une structure de données très utilisée pour représenter des objets. Nous avons présenté différentes méthodes de comparaison, dont les plus connues permettent de décider de l'équivalence (isomorphisme) et de l'inclusion (sous-ismorphisme) de graphes. Ces méthodes sont exactes et ne sont pas tolérantes aux erreurs. Nous avons alors présenté d'autres méthodes plus tolérantes aux erreurs. En particulier une distance entre graphes basée sur la taille de la plus grande partie commune, mais également la distance d'édition de graphes qui est basée sur le nombre d'opérations nécessaires pour obtenir un graphe à partir d'un autre. Nous avons aussi présenté des méthodes de mesures de similarité alternatives basées sur le plongement de graphes dans des espaces vectoriels, que la fonction de plongement soit calculée (plongement de graphes explicites) ou non calculée (plongement de graphes implicites). Ces méthodes permettent d'utiliser les outils existants de classification ou de mesure de distance sur les espaces vectoriels.

Chapitre 3

Cartes généralisées

Sommaire

3.1 Définitions et notations 21
3.2 Isomorphisme et sous-isomorphisme de nG-cartes 24
3.3 Motifs fréquents 26
3.4 Discussion 28

Dans le chapitre précédent nous avons présenté les graphes qui sont utilisés pour représenter la structure de nombreux objets. Nous avons aussi présenté différents types de graphes, dont notamment les graphes plans. Un graphe plan est un graphe planaire plongé dans un plan de sorte que ses arêtes ne s'intersectent pas. Avec ce plongement nous voyons apparaître une nouvelle primitive géométrique de dimension 2, la face. Si les graphes permettent de représenter les sommets et les arêtes, ils ne décrivent cependant pas les faces.

Nous présentons ici les cartes généralisées, ou nG-cartes, qui permettent de représenter les faces en plus des sommets et arêtes dans le cas 2D et, plus généralement, les cellules de toutes dimensions composant un objet en n dimensions (sommets, arêtes, faces, volumes, ...).

3.1 Définitions et notations

Les cartes généralisées sont composées de brins liés deux à deux par des involutions.

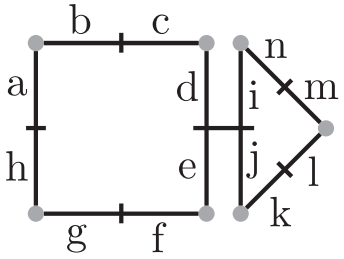
Définition 15 (Involution)
Une involution α sur un ensemble B est une bijection telle que $\forall b \in B, \alpha(\alpha(b)) = b$.

Contrairement aux graphes les cartes généralisées décrivent les cellules (sommets, arêtes et faces, ...) de manière totalement implicite. De plus ces cellules sont définies quelle que soit la dimension et permettent ainsi de manipuler des volumes pour des objets en $3D$, et des cellules de dimension supérieures pour les objets définis en dimensions supérieures.

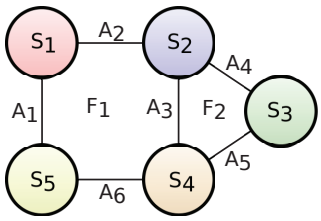
	a	b	c	d	e	f	g	h	i	j	k	l	m	n
α_0	h	c	b	e	d	g	f	a	j	i	l	k	n	m
α_1	b	a	d	c	f	e	h	g	n	k	j	m	l	i
α_2	a	b	c	i	j	f	g	h	d	e	k	l	m	n

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
α_0	h	c	b	e	d	g	f	a	j	k	j	m	l	o	n	i
α_1	b	a	d	c	f	e	h	g	j	i	l	k	n	m	p	o
α_2	a	b	c	o	n	f	g	h	i	j	k	l	m	e	d	p

Définition des involutions α_0, α_1 et α_2 de la 2G-carte.



Représentation graphique de la 2G-carte



Graphe plan correspondant à la 2G-carte

FIGURE 3.1 – *Exemple de $2G$-carte et son graphe plan correspondant. Les brins sont représentés par des segments, ici étiquetés par des lettres. Les brins consécutifs séparés par un petit trait sont 0-cousus (*e.g, *$\alpha_0(b) = c$ et $\alpha_0(c) = b$) ; les brins consécutifs séparés par des ronds gris sont 1-cousus (*e.g., *$\alpha_1(a) = b$ et $\alpha_1(b) = a$) ; les brins parallèles et proches sont 2-cousus (*e.g., *$\alpha_2(d) = i$ et $\alpha_2(i) = d$).*

Définition 16 (Carte généralisée)
Soit $n \geq 0$. Une n-carte généralisée (ou nG-carte) est un tuple $G = (B, \alpha_0, \ldots, \alpha_n)$ où :

1. *B est un ensemble fini de brins ;*
2. *$\forall i : 0 \leq i \leq n$, α_i est une* involution *sur B ;*
3. *$\forall i, j : 0 \leq i < i+2 \leq j \leq n$, $\alpha_i \circ \alpha_j$ est une* involution *sur B.*

Chaque involution α_i lie les brins entre eux. Nous dirons qu'un brin b est i-cousu à un autre brin b' lorsque b et b' sont liés par α_i. Quand un brin b n'est pas i-cousu avec un autre brin (de sorte que $\alpha_i(b) = b$), nous dirons qu'il est i-libre.

Définition 17 (Brin i-libre, i-cousu)
Soient $G = (B, \alpha_0, \ldots, \alpha_n)$ une nG-carte et b un brin de G
- *b est i-cousu à un brin b' de G si $\alpha_i(b) = b'$ et $b \neq b'$.*
- *b est i-libre si $\alpha_i(b) = b$.*

Considérons par exemple la 2G-carte de la Fig. 3.1, le brin a est 0-cousu au brin h, le brin a est 1-cousu au brin b et le brin e est 2-cousu au brin j. Aucun des brins n'est 0-libre ou 1-libre et tous les brins exceptés les brins d, i, e et j sont 2-libres.

Définition 18 (Brin isolé)
Soient $G = (B, \alpha_0, \ldots, \alpha_n)$ une nG-carte et b un brin de G.
b est isolé si $\forall i \in [0, n]$, b est i-libre.

La contrainte 3 de la définition 16 est une contrainte de validité des nG-cartes. Elle garantit leur cohérence. Considérons par exemple la 2G-carte de la Fig. 3.1. Si nous modifions cette 2G-carte en remplaçant $\alpha_2(d) = i$ et $\alpha_2(i) = d$ par $\alpha_2(d) = d$ et $\alpha_2(i) = i$ (i.e., nous 2-décousons les brins i et d) alors la contrainte 3 n'est plus satisfaite puisque

$\alpha_0 \circ \alpha_2$ n'est plus une involution. En effet, $\alpha_0 \circ \alpha_2(d) = e$ et $\alpha_0 \circ \alpha_2(e) = i \neq d$. Pour satisfaire cette contrainte, nous devons également 2-découdre les brins e et j de sorte que $\alpha_2(e) = e$ et $\alpha_2(j) = j$. Cette contrainte de cohérence garantit que si on a 2 arêtes composées de 2 brins, alors l'identification des arêtes identifie la totalité de leurs brins. De manière similaire, en 3D, si nous voulons identifier des faces par α_3, il faut le faire pour tous les brins des 2 faces initiales.

les cartes ainsi définies permettent la subdivision de l'espace de dimension n en cellules de dimensions inférieures ou égales à n de manière implicite. Dans une nG-carte, une i-cellule est définie par un ensemble de brins composé d'un brin de départ et de l'ensemble des brins accessibles depuis ce brin de départ par une combinaison d'involutions. Cette combinaison d'involutions dépend de la dimension de la cellule que nous souhaitons calculer. Pour cela nous utilisons les *orbites*. Soit E un ensemble d'involutions sur B, et b un brin de B. L'orbite $< E > (b)$ est l'ensemble des brins accessibles à partir du brin b en utilisant toutes les involutions de E et toutes les compositions d'involutions de E. Ainsi, une orbite $< \{\alpha_{i1}, \ldots, \alpha_{ik}\} > (b)$ est l'ensemble des brins accessibles en combinant les involutions de $\{\alpha_{i1}, \ldots, \alpha_{ik}\}$ à partir du brin b.

Définition 19 (i-cellule)
Soient $G = (B, \alpha_0, \ldots, \alpha_n)$ une nG-carte, $b \in B$, et $i \in \{0, \ldots, n\}$.
La i-cellule incidente à b, notée $c_i(b)$, est $< \{\alpha_0, \ldots, \alpha_n\} \setminus \{\alpha_i\} > (b)$.

Autrement dit, la i-cellule associée à un brin b est l'ensemble des brins accessibles à partir du brin b en utilisant toutes les involutions excepté α_i. Donnons quelques exemples de i-cellules de la 2G-carte de la Fig. 3.1 en appliquant les orbites correspondantes :

- Exemples de 0-cellules :
  - $S_1 = \{a, b\}$ car $\forall x \in S_1, < \{\alpha_1, \alpha_2\} > (x) = S_1$;
  - $S_2 = \{c, d, i, n\}$;
  - $S_3 = \{m, l\}$.
- Exemples de 1-cellules :
  - $A_1 = \{a, h\}$ car $\forall x \in A_1, < \{\alpha_0, \alpha_2\} > (x) = A_1$;
  - $A_2 = \{b, c\}$;
  - $A_3 = \{d, e, i, j\}$.
- Exemples de 2-cellules :
  - $F_1 = \{a, b, c, d, e, f, g, h\}$ car $\forall x \in F_1, < \{\alpha_0, \alpha_1\} > (x) = F_1$;
  - $F_2 = \{i, j, k, l, m, n\}$.

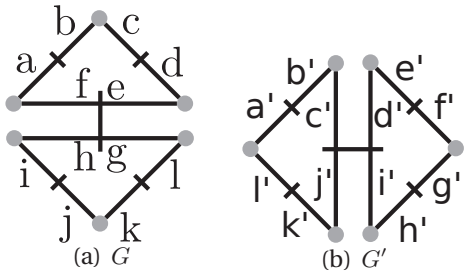
[Lie94] décrit les relations d'adjacence et d'incidence entre les cellules à partir de relations entre leurs ensembles de brins.

Définition 20 (Incidence)
Une i-cellule c_1 est incidente à une j-cellule c_2 si $i \neq j$ et $c_1 \cap c_2 \neq \emptyset$.

Par exemple pour la carte généralisée de la Fig. 3.1, la 0-cellule S_1 est incidente à la 1-cellule A_1 car $S_1 \cap A_1 = \{a\}$, et S_1 est incidente à la 1-cellule A_2 car $S_1 \cap A_2 = \{b\}$. S_1 est aussi incidente à la 2-cellule F_1 car $S_1 \cap F_1 = \{a, b\}$.

Définition 21 (Adjacence)
Deux i-cellules c_1 et c_2 sont adjacentes *s'il existe au moins un brin b_1 de c_1 et un brin b_2 de c_2 tels que $\alpha_i(b_1) = b_2$.*

Par exemple dans la Fig. 3.1 les 0-cellules S_1 et S_2 sont adjacentes car $\alpha_0(b) = c$. Les 1-cellules A_1 et A_2 sont adjacentes car $\alpha_1(a) = b$ et enfin les 2-cellules F_1 et F_2 sont adjacentes car $\alpha_2(d) = i$.



(a) G (b) G'

$brin$	a	b	c	d	e	f	g	h	i	j	k	l
$f(brin)$	k'	l'	a'	b'	c'	j'	d'	i'	h'	g'	f'	e'

(c) fonction isomorphisme f

FIGURE 3.2 – *Exemple de deux nG-cartes isomorphes et de leur fonction isomorphisme f.*

Les relations d'incidence et d'adjacence permettent de décider de l'existence ou non d'une cellule en contact avec une autre cellule. Nous allons aussi définir des notions de degré et de co-degré permettant de décrire le nombre de cellules qui sont effectivement en relation avec une cellule donnée. Dans un graphe, la notion de *degré d'un sommet* nous indique le nombre d'arêtes incidentes à ce sommet. Cette notion a été étendue pour les nG-cartes.

Définition 22 (Degré et co-degré d'une cellule)
Soient une nG-carte G et une i-cellule c_i de G.

- *si $i < n$ alors le degré de c_i, noté $degre(c_i)$, est le nombre de $(i+1)$-cellules incidentes à c_i ;*
- *si $i > 0$ alors le co-degré de c_i, noté $codegre(c_i)$, est le nombre de $(i-1)$-cellules incidentes à c_i.*

Notez que le degré n'est pas défini pour une cellule de dimension n pour une carte généralisée de dimension n, et de manière duale, le co-degré n'est pas défini pour une cellule de dimension 0.
Dans la Fig. 3.1, la 0-cellule S_1 a pour degré 2 car elle est incidente à deux 1-cellules A_1 et A_2. La 0-cellule S_2 a pour degré 3 car elle est incidente aux trois 1-cellules A_2,A_3 et A_4. Toutes les 1-cellules sont de degré 1 excepté la 1-cellule A_3 qui est incidente aux deux 2-cellules, et par conséquent de degré 2.

La 2-cellule F_1 a pour co-degré 4 car elle est incidente à 4 1-cellules. La 2-cellule F_2 a un co-degré de 3 car elle est incidente à trois arêtes. Toutes les 1-cellules ont un co-degré de 2 car elles sont toutes incidentes à 2 0-cellules.

3.2 Isomorphisme et sous-isomorphisme de nG-cartes

L'isomorphisme de nG-cartes permet de décider de l'équivalence de deux nG-cartes et se ramène à la recherche d'une bijection entre leurs brins telle que toutes les involutions soient conservées.

Définition 23 (Isomorphisme [Lie94])
Deux nG-cartes $G = (B, \alpha_0, \ldots, \alpha_n)$ et $G' = (B', \alpha'_0, \ldots, \alpha'_n)$ sont isomorphes (noté $G \simeq G'$) s'il existe une bijection $f : B \to B'$ vérifiant : $\forall i \in [0,n],\ \forall b \in B,\ f(\alpha_i(b)) = \alpha'_i(f(b))$.

La Fig. 3.2 met en évidence l'isomorphisme de deux 2G-cartes.

[Cor73, Lie91] ont défini la notion d'isomorphisme entre 2 cartes combinatoires, duquel découle directement un algorithme polynomial. Pour cela, un ordre de parcours est choisi pour les involutions (par exemple α_0, α_1 puis α_2) puis à partir d'un brin de G, et pour chaque brin de G' les brins de G sont parcourus selon l'ordre fixé, reproduit à l'identique sur G'. Nous obtenons ainsi une mise en correspondance des brins des deux nG-cartes. Si pour cette mise en correspondance toutes les coutures entre brins sont préservées, alors les nG-cartes sont isomorphes. Sinon il faut recommencer un nouveau parcours à partir d'un autre brin de G, jusqu'à soit trouver une mise en correspondance respectant les coutures (de sorte que les nG-cartes soient isomorphes), soit montrer qu'aucune mise en correspondance ne respecte les coutures (de sorte que les nG-cartes ne soient pas isomorphes).
Par exemple sur la Fig. 3.2, nous choisissons l'ordre de parcours α_0, α_1 puis α_2 ainsi qu'un brin de G et un brin de G'. Nous choisissons donc arbitrairement d et d', associons d à d' et commençons à parcourir les deux cartes dans l'ordre fixé. Nous débutons par d et nous découvrons par α_0 le brin c, dans G' c'est i' que nous découvrons car $\alpha_0(d') = i'$. Nous associons c à i' puis nous continuons avec α_1 où nous découvrons e et e' que nous associons. Enfin par α_2 nous ne découvrons pas de nouveaux brins dans G alors que nous en découvrons un dans G' car $\alpha_2(d') = c'$. Ainsi, en partant de d et d', nous ne trouvons pas d'isomorphisme. Nous changeons donc de brin dans G, et nous choisissons arbitrairement e que nous associons à d'. Cette fois, le parcours se fait sans interruption et calcule un isomorphisme. La complexité en temps de cet algorithme est de $O(n|B|^2)$ dans le pire des cas (si les nG-cartes ne sont pas isomorphes).

La relation d'équivalence est la comparaison la plus basique entre nG-cartes, mais elle ne permet pas de décrire les relations d'inclusion. Pour cela, [DSd+11] définit le sous-isomorphisme de nG-cartes. Une nG-carte G est une sous-carte de G' si les brins de G sont un sous-ensemble des brins de G' et si toutes les involutions entre les brins de G sont conservées.

L'isomorphisme de sous-carte est défini à partir de la notion de sous-carte.

Définition 24 (Sous-carte)
Soient $G = (B, \alpha_0, \ldots, \alpha_n)$ et $G' = (B', \alpha'_0, \ldots, \alpha'_n)$ deux nG-cartes.
G est une sous-carte de G' si $B \subseteq B'$ et si $\forall b \in B, \forall i \in [0, n]$
- *$\alpha'_i(b) \in B \Rightarrow \alpha_i(b) = \alpha'_i(b)$;*
- *$\alpha'_i(b) \notin B \Rightarrow \alpha_i(b) = b$.*

Définition 25 (Sous-isomorphisme)
Soient $G = (B, \alpha_0, \ldots, \alpha_n)$ et $G' = (B', \alpha'_0, \ldots, \alpha'_n)$ deux nG-cartes. G est sous-isomorphe à G', noté $G \sqsubseteq G'$ s'il existe une sous-carte de G' qui soit isomorphe à G.

Notez que à l'instar de l'isomorphisme de sous-graphes induit, où toutes les arêtes doivent être conservées entre les sommets du sous-graphe, ici toutes les involutions doivent être conservées entre les brins qui font partie de la sous-carte. Dans la Fig. 3.3 nous illustrons le concept de sous-cartes.

La complexité du problème de la recherche de sous-isomorphisme de nG-carte dépend de la connexité des nG-cartes [DLHJS+]. Une nG-carte $G = (B, \alpha_0, \ldots, \alpha_n)$ est dite connexe si pour tout $b \in B, < \alpha_0, \ldots, \alpha_n > (b) = B$, autrement dit, si tout brin est accessible à partir de b. Considérons le problème de décider si $G' \sqsubseteq G$. Dans le cas où la nG-carte G' est connexe le problème de recherche de sous-isomorphisme est polynomial (que G soit connexe ou non). Pour rechercher cet isomorphisme, nous choisissons un brin de G', et nous essayons d'associer ce brin avec chaque brin de G. Nous réalisons

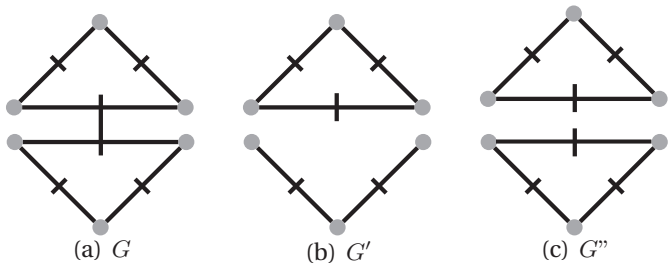


FIGURE 3.3 – *G' est une sous-carte de G. Il existe une injection entre ses brins et les brins de G telle que toutes les involutions entre deux brins de G' dans G soient soient conservées. De la même manière G' est une sous-carte de G''. Il existe une injection entre les brins de G' et ceux de G''. Par contre il n'existe pas une injection entre les brins de G'' et ceux de G conservant les involutions de G, en effet, si l'on associe les deux faces triangulaires de G'' aux deux faces triangulaires de G, les 2-coutures de G entre les deux faces ne seront pas conservées dans G''. G'' n'est dont pas une sous-carte de G.*

ensuite un parcours de brins de G' de manière similaire à l'isomorphisme, en vérifiant que ce même parcours existe dans G à partir du brin choisi. Nous vérifions aussi que les coutures de G' sont bien présentes dans G, et qu'il n'existe pas de coutures entre deux brins de G ayant une image par la fonction de sous-isomorphisme qui ne sont pas présentes dans G'. La complexité en temps de cet algorithme est $O(n|B'||\dot{B}|)$. Dans le cas où G' n'est pas connexe, le problème devient NP-complet. En effet le problème planar 4-3-SAT peut être réduit en temps polynomial au problème d'isomorphisme de sous-cartes dans ce cas. Planar 4-3-SAT est un problème de satisfiabilité d'une formule booléenne sous forme clausale dont toutes les clauses ont exactement 3 littéraux et dont chaque variable appartient à au plus 4 clauses, tel que le graphe dont les sommets sont les variables et les arêtes la relation de co-occurence de 2 variables dans une clause, est planaire.

3.3 Motifs fréquents

[GDS11] introduit un algorithme pour rechercher des motifs (sous-cartes) fréquents dans une base de cartes combinatoires (les cartes combinatoires étant des cas particuliers des cartes généralisées). Soient $\{G_0, \ldots, G_k\}$ une base de cartes combinatoires, $\theta \in [0,1]$ et $\phi \in [0, +\infty]$. L'algorithme retourne l'ensemble M de toutes les cartes combinatoires qui sont présentes au moins ϕ fois dans au moins $\theta.k$ cartes de la base. Les cartes combinatoires de M sont appelées des motifs fréquents. Pour construire cet ensemble de motifs, nous utilisons une forme canonique des cartes combinatoires appelée signature. Cette signature permet en complexité linéaire de décider de l'isomorphisme de deux cartes. Pour générer l'ensemble des motifs fréquents, l'algorithme construit les motifs composés d'une seule face, puis il combine ces motifs avec les motifs d'une seule face pour construire des motifs de 2 faces et plus. Comme illustré dans la figure Fig. 3.4 il est ainsi possible de générer l'ensemble des motifs fréquents d'une base de données de cartes combinatoires en temps polynomial incrémental, i.e., la complexité en temps pour générer chaque motif est polynomiale. En revanche, le nombre de motifs peut être très grand (exponentiel) et dépend fortement des seuils de fréquence ϕ et θ de sorte qu'il n'est pas possible de générer l'ensemble des motifs communs (sans contrainte de fréquences) en un temps raisonnable.

Cet algorithme de recherche de motifs fréquents peut être utilisé pour plonger un

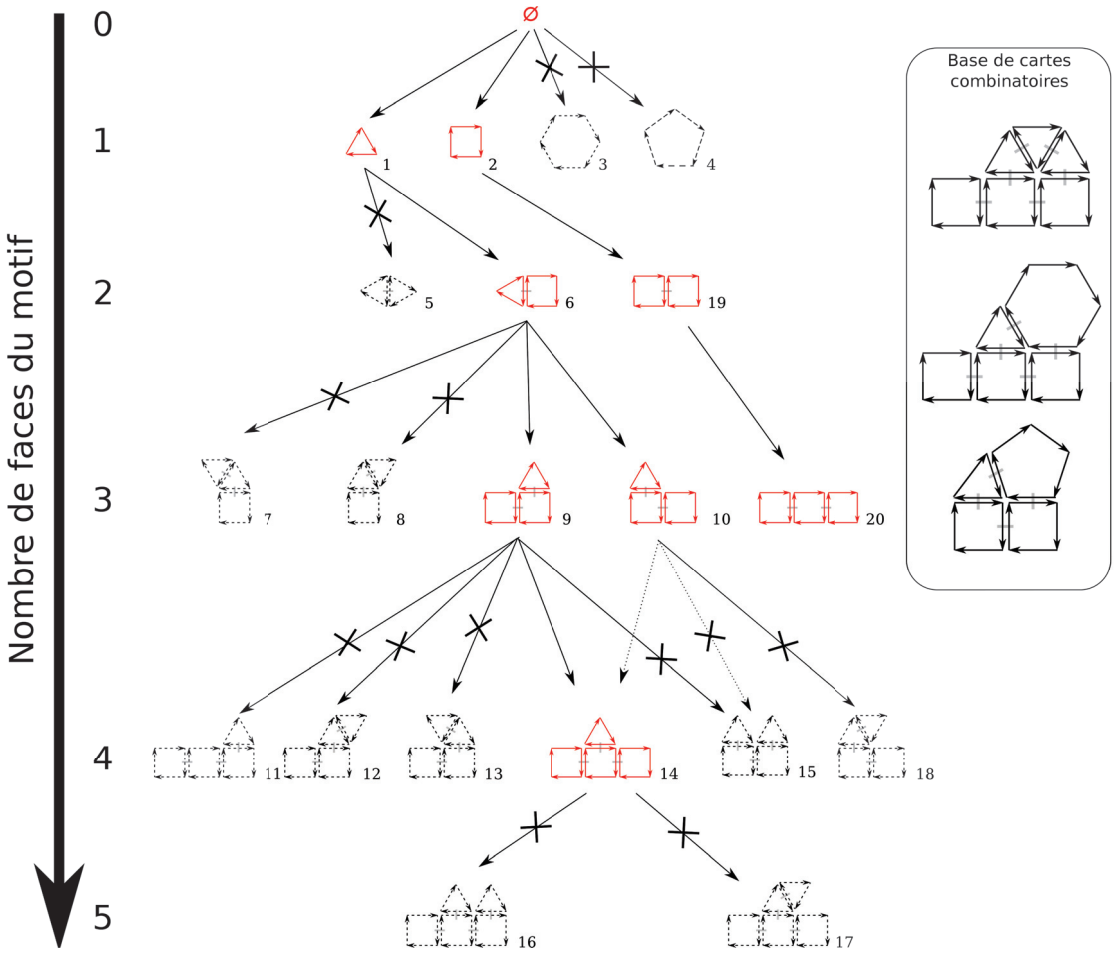


FIGURE 3.4 – *Illustration de la construction d'un ensemble de motifs fréquents pour une base de cartes combinatoires donnée. À chaque étape nous ajoutons aux motifs fréquents (rouges) les motifs de taille 1. Nous vérifions ensuite leur fréquence, et nous itérons successivement sur tous les niveaux. Nous considèrons ici qu'un motif est fréquent s'il apparait au moins 1 fois dans $\frac{2}{3}$ des cartes de la base.*

ensemble de cartes combinatoires dans un espace vectoriel. Considérons l'ensemble $\{G_0, \ldots, G_k\}$ des cartes de la base, et $\{G'_0, \ldots, G'_l\}$ les sous-cartes fréquentes. Chaque carte combinatoire $G_i \in \{G_0, \ldots, G_k\}$ est représentée par un vecteur $(\omega_0, \ldots, \omega_l)$ où ω_j est le nombre d'apparitions du motif G'_j dans la carte G_i. Dans cet espace vectoriel, nous pouvons utiliser la distance euclidienne comme mesure de similarité. Cette méthode se révèle très efficace pour la classification de cartes combinatoires, mais ne peut pas être adaptée facilement pour comparer deux nG-cartes au sens où les motifs et donc la mesure de similarité sont dépendants de l'ensemble des cartes comparées. Pour mesurer la similarité entre deux cartes, il faut donc tout d'abord choisir les motifs fréquents pertinents avec une base d'apprentissage.

3.4 Discussion

Nous avons présenté dans cette section les cartes généralisées qui permettent de représenter des objets subdivisés dans des espaces en n dimensions. Une carte généralisée est composée de brins qui sont mis en relation par des involutions. Les cellules de la subdivision ne sont pas décrites explicitement mais implicitement au moyen d'orbites. Nous avons aussi présenté l'ensemble des méthodes permettant de comparer des nG-cartes : l'isomorphisme permettant de décider de l'équivalence de deux nG-cartes ; le sous-isomorphisme permettant de décider de la relation d'inclusion entre nG-cartes. Ces deux méthodes ne sont absolument pas tolérantes aux erreurs, et la moindre différence les mettront en défaut. Ensuite, nous avons présenté une méthode basée sur le dénombrement de motifs (sous-cartes) communs entre deux nG-cartes. Il est ainsi possible de représenter une nG-carte par un vecteur énumérant le nombre d'apparition de chacun des motifs. La complexité de cette méthode dépend du nombre de motifs existants, et il n'est généralement pas possible de dénombrer tous les motifs d'une nG-carte en temps raisonnable. La pertinence de la mesure de similarité dépend donc du choix des motifs fréquents dénombrés. Pour cela [Gos11] utilise une base d'apprentissage sur laquelle les motifs fréquents pertinents sont recherchés. Nous allons maintenant proposer une mesure de similarité permettant d'évaluer si deux nG-cartes sont similaires.

Chapitre 4

Définition de distances entre cartes généralisées

Sommaire

4.1 Sous-cartes induites et sous-cartes partielles 29
4.2 Distance basée sur la plus grande sous-carte commune 32
4.3 Distance d'édition entre nG-cartes 38
4.4 Relation entre d_{mcs} et d_{ed} 41
4.5 nG-cartes étiquetées et distance d'édition 48
4.6 Discussion 50

Dans ce chapitre, nous nous inspirons d'un certain nombre de travaux existants sur les graphes pour proposer de nouvelles mesures de similarité sur les nG-cartes qui soient tolérantes aux erreurs. La Section 4.1 commence tout d'abord par définir les notions de sous-cartes induites et partielles nécessaires pour ensuite proposer une distance basée sur la plus grande sous-carte commune dans la Section 4.2. Dans un second temps, nous proposons une distance d'édition (Section 4.3), puis nous mettons en évidence un lien entre cette distance d'édition et les plus grandes sous-cartes communes. Enfin nous proposons d'intégrer de l'information dans la Section 4.5 en ajoutant des étiquettes aux brins et en adaptant la distance d'édition en conséquence.

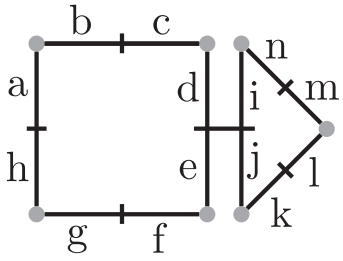
Dans ce chapitre G, G' et G'' désigneront respectivement les nG-cartes $(B, \alpha_0, \ldots, \alpha_n)$, $(B', \alpha'_0, \ldots, \alpha'_n)$ et $(B'', \alpha''_0, \ldots, \alpha''_n)$.

4.1 Sous-cartes induites et sous-cartes partielles

Afin de simplifier de futures définitions et théorèmes, nous définissons la notion de *coutures* associées à un ensemble de brins.

Définition 26 (Coutures associées à un ensemble de brins)
Soient G une nG-carte et $E \subseteq B$ un sous-ensemble de brins de G. L'ensemble des coutures associées à E dans G est :
$coutures_G(E) = \{(b, i, \alpha_i(b)) | b \in E, i \in [0, n], \alpha_i(b) \in E, \alpha_i(b) \neq b\}$.

Autrement dit, l'ensemble des coutures d'un sous-ensemble de brins E dans une nG-carte est l'ensemble des triplets (b, i, b') avec b et b' dans E et tels que b soit i-cousu à b'.



(a) $G = (B, \alpha_0, \ldots, \alpha_n)$

$E \subseteq B$	$coutures_G(E)$
$\{k\}$	$\emptyset$
$\{a,b,c\}$	$\{(a,1,b),(b,1,a),(b,0,c),(c,0,b)\}$
$\{e,j,k)$	$\{(e,2,j),(j,2,e),(j,1,k),(k,1,j)\}$
$\{d,i,j,e\}$	$\{(d,2,i),(i,2,d),(d,0,e),(e,0,d),(i,0,j),(j,0,i),(e,2,j),(j,2,e)\}$

FIGURE 4.1 – *Exemple d'une* 2*G-carte, et d'ensembles de coutures.*

Notez que si le triplet (b,i,b') fait partie d'un ensemble de coutures, alors le triplet (b',i,b) fait aussi partie de cet ensemble. Cela est inhérent au fait que α_i est une involution et donc si $\alpha_i(b) = b'$ alors $\alpha_i(b') = b$. La Fig. 4.1 illustre cette notion de coutures associées à un ensemble de brins.

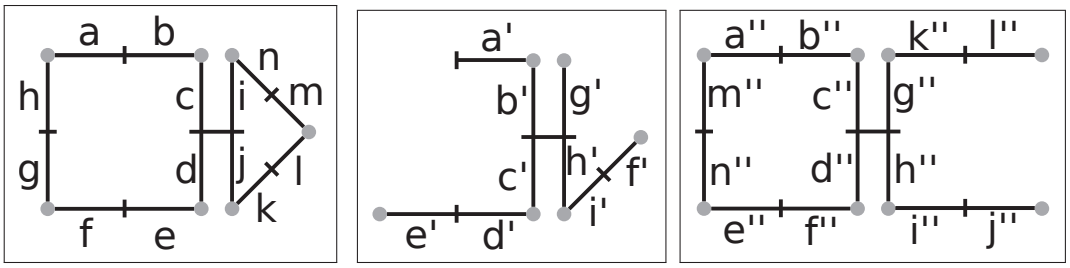
Une nG-carte peut maintenant être décrite comme un ensemble de brins et un ensemble de coutures. La suppression d'une couture (b,i,b') et de son symétrique (b',i,b) dans une nG-carte est équivalent à rendre i-libre (découdre) les brins b et b' (c'est-à-dire remplacer $\alpha_i(b) = b'$ et $\alpha_i(b') = b$ par $\alpha_i(b) = b$ et $\alpha_i(b') = b'$). L'ajout d'une couture (b,i,b') et de son symétrique (b',i,b) dans la nG-carte engendrera la i-couture de b et b' (c'est-à-dire remplacera $\alpha_i(b) = b$ et $\alpha_i(b') = b'$ par $\alpha_i(b) = b'$ et $\alpha_i(b') = b$).

Nous proposons de mesurer la similarité entre nG-cartes, par rapport à leur plus grande sous-carte commune. De la même façon que la notion de plus grand sous-graphe commun peut considérer des sous-graphes induits (conservant toutes les arêtes entre les sommets des sous-graphes) ou des sous-graphes partiels (ne conservant pas nécessairement toutes les arêtes), nous proposons de considérer 2 types de sous-cartes : des sous-cartes induites par les brins (dites sous-cartes *induites*) et des sous-cartes induites par les coutures (dites sous-cartes *partielles*). L'isomorphisme de sous-carte introduit dans [DSd+11] et rappelé dans la section précédente (définition 24) correspond en fait à ce que nous appellerons isomorphisme de sous-cartes induites dans la suite. C'est dans l'objectif de mettre en évidence la différence entre les deux types de sous-cartes (induites et partielles) que nous proposons une réécriture basée sur les coutures. La sous-carte induite est une sous-carte qui, pour son sous-ensemble de brins conserve la totalité des coutures les mettant en relation.

Définition 27 (Sous-carte induite)
G' est une sous-carte induite de G si $B' \subseteq B$ et $coutures_{G'}(B') = coutures_G(B')$.

Définition 28 (Isomorphisme de sous-carte induite)
Il existe un isomorphisme de sous-carte induite de G' vers G, noté $G' \sqsubseteq^i G$ s'il existe une sous-carte induite de G qui est isomorphe à G'.

Notez que s'il existe un isomorphisme de sous-carte induite de G' vers G, alors il existe une injection $f : B' \to B$ appelée *fonction de sous-isomorphisme induit* telle que $\forall b' \in B'$ et $\forall i \in [0,n]$:



(a) G (b) G' (c) G''

brin	a'	b'	c'	d'	e'	f'	g'	h'	i'
f(brin)	b	c	d	e	f	l	i	j	k

(d) fonction de sous-isomorphisme induite $G' \to G$

brin	a"	b"	c"	d"	e"	f"	g"	h"	i"	j"	k"	l"	m"	n"
f'(brin)	a	b	c	d	f	e	i	j	k	l	n	m	h	g

(e) fonction de sous-isomorphisme partiel $G'' \to G$

FIGURE 4.2 – *Exemple de sous-carte induite et de sous-carte partielle. $G' \sqsubseteq^i G$, et f est la fonction de sous-isomorphisme induit mettant en relation les brins de G' et les brins de G telle que toutes les coutures soient conservées. $G'' \sqsubseteq^p G$, et f' est la fonction de sous-isomorphisme partiel mettant en relation les brins de G'' avec les brins de G. Notez que les coutures $(m,1,l)$ et $(l,1,m)$ ne sont pas conservées : G'' n'est pas une sous-carte induite de G.*

– si b' est i-cousu, alors $f(\alpha'_i(b')) = \alpha_i(f(b'))$;
– si b' est i-libre, alors soit $f(b')$ est i-libre, soit $f(b')$ est i-cousu avec un brin qui n'est pas apparié par f à un autre brin de B', *i.e.*, $\forall b_k \in B', f(b_k) \neq \alpha_i(f(b'))$.

La Fig. 4.2 illustre le concept de sous-carte induite : G' est isomorphe à une sous-carte induite de G car il existe une fonction f mettant en relation les brins de G' et les brins de G telle que toutes leurs coutures soient conservées.

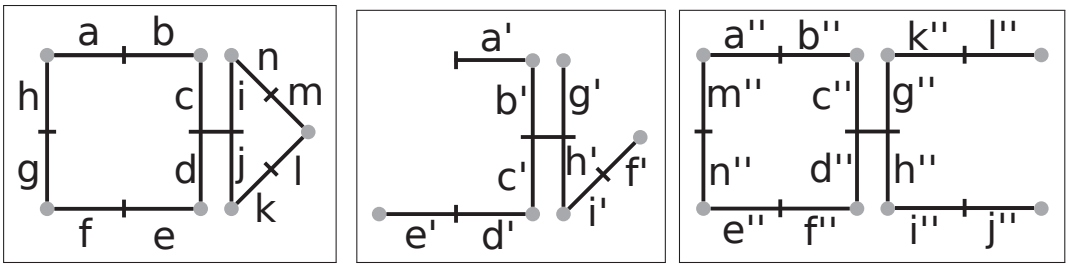
La définition d'une sous-carte induite ne considère pas une nG-carte à laquelle nous avons simplement supprimé une couture comme sous-carte de la nG-carte initiale. Par exemple dans la Fig. 4.2 $G'' \not\sqsubseteq^i G$ alors que G'' peut être rendue isomorphe à G en 1-cousant les brins l'' et j''. Ces deux nG-cartes sont cependant très proches. Dans un souci de flexibilité et de tolérance à l'erreur nous définissons un second type de sous-carte pour laquelle toutes les coutures ne seront plus obligatoirement conservées. Ce nouveau type de sous-carte est appelé *sous-carte partielle*, par analogie aux sous-graphes partiels pour lesquels les arêtes ne sont pas obligatoirement conservées.

Définition 29 (Sous-carte partielle)
G' est une sous-carte partielle de G si $B' \subseteq B$ et $coutures_{G'}(B') \subseteq coutures_G(B')$.

L'isomorphisme de sous-carte est étendu dans le cas partiel de manière directe.

Définition 30 (Isomorphisme de sous-carte partielle)
Il y a un isomorphisme de sous-carte partielle de G' vers G, noté $G' \sqsubseteq^p G$ s'il existe une sous-carte partielle de G qui est isomorphe à G'.

Notons que $G' \sqsubseteq^i G \Rightarrow G' \sqsubseteq^p G$ que nous déduisons trivialement des définitions. Nous pouvons aussi remarquer que si $G' \sqsubseteq^p G$ alors il existe une fonction injective $f : B' \to B$, appelée *fonction de sous-isomorphisme partiel*, telle que $\forall b' \in B'$ et $\forall i \in [0,n]$: si b' est i-cousu, alors $f(\alpha'_i(b')) = \alpha_i(f(b'))$.



(a) G (b) G' (c) G''

Carte	$taille_{1,0}$	$taille_{0,1}$	$taille_{1,1}$	$taille_{2,1}$	$taille_{1,2}$
G	14	32	46	60	78
G'	9	18	27	36	45
G''	14	30	44	58	74

(d) Exemples de tailles de nG-cartes

FIGURE 4.3 – *Exemples d'instanciations de la taille paramétrée sur les trois 2G-cartes G, G' et G''. Notez que $taille_{1,0}(G) = taille_{1,0}(G'')$ alors que $G \not\simeq G''$. Pour les autres, la pondération permet de changer l'importance des coutures par rapport aux brins.*

Afin d'illustrer le concept de sous-cartes partielles, considérons les 2G-cartes de la Fig. 4.2. Nous avons $G' \sqsubseteq^p G$ et $G' \sqsubseteq^p G''$ car $G' \sqsubseteq^i G$ et $G' \sqsubseteq^i G''$. Nous avons aussi $G'' \sqsubseteq^p G$. En effet, la suppression des coutures $(m,1,l)$ et $(l,1,m)$ de G modifie la 2G-carte G en une 2G-carte isomorphe à G''. Par contre G'' n'est pas une sous-carte induite de G car les involutions $(m,1,l)$ et $(l,1,m)$ ne sont pas retrouvées.

Tout au long du manuscrit, $*$ dénotera p ou i (i.e., $* \in \{p,i\}$) de sorte que $G' \sqsubseteq^* G$ signifiera soit un isomorphisme de sous-carte induite ($G' \sqsubseteq^i G$) ou soit un isomorphisme de sous-carte partielle ($G' \sqsubseteq^p G$).

4.2 Distance basée sur la plus grande sous-carte commune

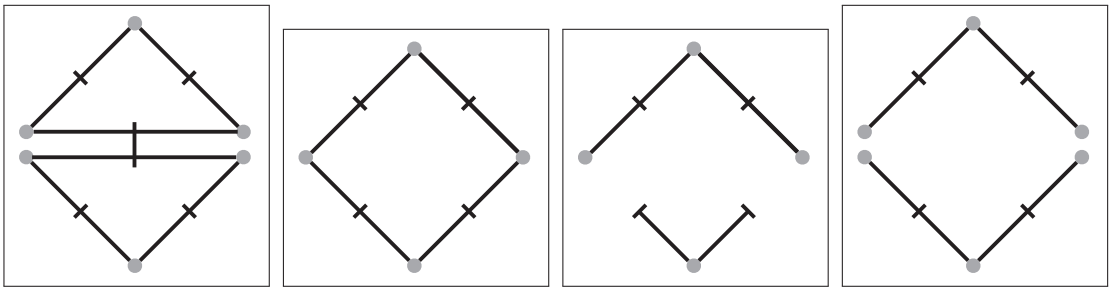
Taille d'une nG-carte

La taille d'une nG-carte est une notion primordiale pour quantifier la partie commune entre deux nG-cartes. Les nG-cartes étant composées de brins mis en relation par des involutions, il semble normal de considérer une taille basée sur le nombre de brins. Seulement, l'information contenue dans une nG-carte n'est pas seulement dépendante de son nombre de brins, mais aussi de la façon dont ils sont cousus. Pour cela, nous proposons une taille paramétrée permettant de pondérer l'importance des brins par rapport aux coutures.

Définition 31 (Taille paramétrée d'une nG-carte)
Étant donné $(\omega_1,\omega_2) \in \mathbb{R}^{+2}$ tel que $(\omega_1,\omega_2) \neq (0,0)$, la taille d'une nG-carte G est

$$taille_{\omega_1,\omega_2}(G) = \omega_1.|B| + \omega_2.|coutures_G(B)|.$$

La Fig. 4.3 illustre différentes instanciations de cette taille paramétrée sur quelques exemples : $taille_{1,0}(G)$ est égal à 14 le nombre de brins de G, et $taille_{0,1}(G)$ est égal à 32 le nombre de coutures de G. Notez que selon les valeurs choisies pour ω_1 et ω_2, nous pouvons avoir 2 nG-cartes G_1 et G_2 non isomorphes, l'une étant sous-carte de l'autre, mais pour lesquelles $taille_{\omega_1,\omega_2}(G_1) = taille_{\omega_1,\omega_2}(G_2)$. C'est le cas par exemple pour les



(a) G (b) G' (c) G'' (d) G'''

FIGURE 4.4 – *Exemples de plus grandes sous-cartes communes. G'' est une plus grande sous-carte commune induite de G et G' quand $\omega_1 = \omega_2 = 1$, i.e., $mcs^i_{1,1}(G,G') \simeq G''$, et $taille_{1,1}(G'') = 6+8$. G''' est une plus grande sous-carte commune partielle de G et G' quand $\omega_1 = \omega_2 = 1$, i.e., $mcs^p_{1,1}(G,G') \simeq G'''$, et $taille_{1,1}(G''') = 8+12$.*

2G-cartes G et G'' de la Fig. 4.3, quand $\omega_2 = 0$. Il convient donc d'utiliser une pondération et un type de sous-carte adéquat en fonction de ce que nous souhaitons comparer. Nous mettrons en évidence dans les définitions et les illustrations des futures distances, différentes pondérations de taille.

Plus grande sous-carte commune

Nous pouvons maintenant définir la plus grande sous-carte commune à deux nG-cartes.

Définition 32 (Plus grande sous-carte commune)
Soient $(\omega_1,\omega_2) \in \mathbb{R}^{+2}$ tel que $(\omega_1,\omega_2) \neq (0,0)$, $* \in \{i,p\}$ et 2 nG-cartes G et G'. Une plus grande sous-carte commune de G et G', notée $mcs^*_{\omega_1,\omega_2}(G,G')$, est une nG-carte telle que :

- $mcs^*_{\omega_1,\omega_2}(G,G') \sqsubseteq^* G$;
- $mcs^*_{\omega_1,\omega_2}(G,G') \sqsubseteq^* G'$;
- $taille_{\omega_1,\omega_2}(mcs^*_{\omega_1,\omega_2}(G,G'))$ *est maximale, i.e. pour toute nG-carte G'' telle que $G'' \sqsubseteq^* G$ et $G'' \sqsubseteq^* G'$, $taille_{\omega_1,\omega_2}(mcs^*_{\omega_1,\omega_2}(G,G')) \geq taille_{\omega_1,\omega_2}(G'')$.*

$mcs^i_{\omega_1,\omega_2}(G,G')$ est appelée une plus grande sous-carte commune induite, et $mcs^p_{\omega_1,\omega_2}(G,G')$ une plus grande sous-carte commune partielle. Notez que cette plus grande-sous carte commune n'est pas unique : il est tout à fait possible qu'il existe une sous-carte commune ayant la même taille que $mcs^i_{\omega_1,\omega_2}(G,G')$.

La Fig. 4.4 donne quelques exemples de plus grandes sous-cartes communes.

Notez que de la définition nous déduisons directement que toute plus grande sous-carte commune de G et G' est également une plus grande sous-carte commune de G' et G. L'écriture $mcs^*_{\omega_1,\omega_2}(G,G')$ est donc équivalente à $mcs^*_{\omega_1,\omega_2}(G',G)$.

Complexité du problème de recherche d'une plus grande sous-carte commune

Nous avons donné dans la Section 3.2 la complexité du problème d'isomorphisme de sous-carte induite : le problème de décider si $G \sqsubseteq^i G'$ est polynomial dans le cas où G est connexe et NP-complet dans le cas où G n'est pas connexe. Ce résultat de NP-complétude dans le cas où G n'est pas connexe s'étend directement au cas de l'isomorphisme de sous-carte partielle. La recherche de la plus grande sous-carte commune

est donc un problème NP-difficile si on n'impose pas que la sous-carte commune soit connexe (ce qui est le cas de la définition 32).

La complexité du problème de recherche de plus grande sous-carte commune dans le cas où nous imposons que la plus grande-sous-carte commune soit connexe reste ouvert : si nous n'avons pas trouvé d'algorithme polynomial pour résoudre ce problème nous n'avons pas non plus trouvé de réduction permettant d'affirmer que le problème est NP-complet.

Définition de la distance

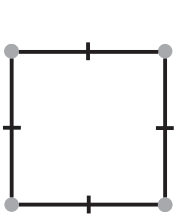
Nous avons proposé une taille de nG-carte pour laquelle l'importance des brins et des coutures peut être pondérée. La combinaison de cette taille et d'un type de sous-carte (induite ou partielle) génère des plus grandes sous-cartes communes différentes. Afin de préserver cette généricité nous définissons maintenant une distance paramétrable. Cette distance est inspirée de la distance de Bunke et Shearer dans [BS98] qui est le rapport entre la taille du plus grand sous-graphe commun et la plus grande des tailles des deux graphes. Dans le cas des nG-cartes, notre distance est définie comme le rapport entre la taille de la plus grande partie commune et la taille de la plus grande des deux nG-cartes. Elle est paramétrée par ω_1 et ω_2 qui pondèrent l'importance des brins et des coutures et par le type de sous-carte (induit ou partiel).

Définition 33 (Distance d_{mcs} paramétrée entre deux nG-cartes)
Soient $(\omega_1, \omega_2) \in \mathbb{R}^{+2}$ *tel que* $(\omega_1, \omega_2) \neq (0,0)$ *et* $* \in \{i, p\}$.
La distance d_{mcs} *entre* G *et* G' *est définie par :*

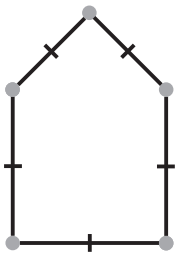
$$d^{*,\omega_1,\omega_2}_{mcs}(G,G') = 1 - \frac{taille_{\omega_1,\omega_2}(mcs^{*}_{\omega_1,\omega_2}(G,G'))}{max(taille_{\omega_1,\omega_2}(G), taille_{\omega_1,\omega_2}(G'))}$$

La Fig. 4.5 donne quelques exemples de distances. Pour le premier couple de 2G-cartes G et G', notez que la valeur de la distance reste proche de 0.3 quel que soit le type de sous-carte considéré (induit ou partiel), et pour les différentes valeurs de ω_1 et ω_2 considérées. Pour la seconde paire de 2G-cartes, notez deux résultats intéressants. Tout d'abord, $d^{p,1,0}_{mcs}(G'', G''') = 0$ sans pour autant que les 2G-cartes soient isomorphes. Ce résultat montre que la distance ne permet pas de définir un espace métrique dans le cas partiel quand $\omega_1 = 1$ et $\omega_2 = 0$. Ensuite, notez que les 2G-cartes sont très proches (les deux faces ont juste été décousues), et pourtant la distance utilisant la sous-carte induite reste élevée (0.17, 0.36 et 0.27), alors que la distance utilisant les sous-cartes partielles qui prend en compte les coutures et les brins met mieux en évidence la proximité des 2G-cartes (0.077).

Montrons que notre définition de distance permet de définir un espace métrique quand $* = p$ et ω_1 et ω_2 sont non nuls ou quand $* = i$ et ω_1 est non nul.



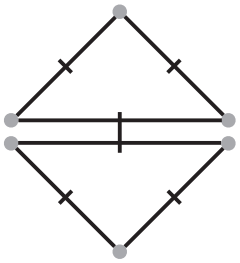
(a) G



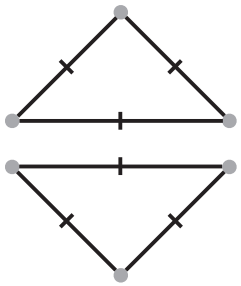
(b) G'

$d_{mcs}^{i,1,0}(G,G')$	0.3
$d_{mcs}^{i,0,1}(G,G')$	0.4
$d_{mcs}^{i,1,1}(G,G')$	0.367
$d_{mcs}^{p,1,0}(G,G')$	0.2
$d_{mcs}^{p,0,1}(G,G')$	0.3
$d_{mcs}^{p,1,1}(G,G')$	0.267

(c) Distances



(d) G''



(e) G'''

$d_{mcs}^{i,1,0}(G'',G''')$	0.17
$d_{mcs}^{i,0,1}(G'',G''')$	0.357
$d_{mcs}^{i,1,1}(G'',G''')$	0.269
$d_{mcs}^{p,1,0}(G'',G''')$	0
$d_{mcs}^{p,0,1}(G'',G''')$	0.143
$d_{mcs}^{p,1,1}(G'',G''')$	0.077

(f) Distances

FIGURE 4.5 – *Illustration de distances entre deux paires de* 2*G-cartes* G *et* G' *puis* G'' *et* G'''*. Entre* G *et* G'*, quelques soient les paramètres utilisés la distance reste très stable (proche de* 0.3*). Entre* G'' *et* G'''*, le fait de considérer la sous-carte partielle et de tenir compte des brins et des coutures pour la distance représente mieux la proximité des deux* 2*G-cartes (*0.077*). Par contre le poids sur les coutures ne doit pas être nul, sinon la distance entre les deux* 2*G-cartes est* 0 *sans pour autant qu'elles soient isomorphes.*

Théorème 1
Soient $* \in \{i,p\}, \omega_1 \geq 0, \omega_2 \geq 0$*. La distance* $d_{mcs}^{*,\omega_1,\omega_2}$ *est une métrique sur l'ensemble* $\mathcal{G}$ *de toutes les* n*G-cartes si (*$* = p, \omega_1 \neq 0$ *et* $\omega_2 \neq 0$*) ou (*$* = i$ *et* $\omega_1 \neq 0$*).*

Démonstration. Pour montrer que $d_{mcs}^{*,\omega_1,\omega_2}$ est une métrique nous devons montrer les trois propriétés suivantes :

1. Identité des indiscernables :
   $\forall G_1, G_2 \in \mathcal{G}, d_{mcs}^{*,\omega_1,\omega_2}(G_1,G_2) = 0$ ssi G_1 et G_2 sont isomorphes ;
2. Symétrie :
   $\forall G_1, G_2 \in \mathcal{G}, d_{mcs}^{*,\omega_1,\omega_2}(G_1,G_2) = d_{mcs}^{*,\omega_1,\omega_2}(G_2,G_1)$;
3. Inégalité triangulaire :
   $\forall G_1, G_2, G_3 \in \mathcal{G}, d_{mcs}^{*,\omega_1,\omega_2}(G_1,G_3) \leq d_{mcs}^{*,\omega_1,\omega_2}(G_1,G_2) + d_{mcs}^{*,\omega_1,\omega_2}(G_2,G_3)$.

La propriété 2 est une conséquence directe de la définition 33.

Preuve de la propriété 1 Supposons $d_{mcs}^{*,\omega_1,\omega_2} = 0$. En utilisant la définition 33 nous obtenons : $taille_{\omega_1,\omega_2}(mcs_{\omega_1,\omega_2}^*(G_1,G_2)) = max(taille_{\omega_1,\omega_2}(G_1), taille_{\omega_1,\omega_2}(G_2))$. Avec la définition 32 nous savons aussi que $mcs_{\omega_1,\omega_2}^*(G_1,G_2) \sqsubseteq^* G_1$ et que $mcs_{\omega_1,\omega_2}^*(G_1,G_2) \sqsubseteq^* G_2$. Il suit que $taille_{\omega_1,\omega_2}(mcs_{\omega_1,\omega_2}^*(G_1,G_2)) = taille_{\omega_1,\omega_2}(G_1) = taille_{\omega_1,\omega_2}(G_2)$.

$mcs_{\omega_1,\omega_2}^*(G_1,G_2) \sqsubseteq^* G_1$ et $mcs_{\omega_1,\omega_2}^*(G_1,G_2) \sqsubseteq^* G_2$ donc il existe une injection entre $mcs_{\omega_1,\omega_2}^*(G_1,G_2)$ et une sous-carte de G_1 et aussi une injection de $mcs_{\omega_1,\omega_2}^*(G_1,G_2)$ vers une sous-carte de G_2. De plus nous avons montré que les tailles sont égales, et comme $mcs_{\omega_1,\omega_2}^*(G_1,G_2) \sqsubseteq^* G_1$ et $mcs_{\omega_1,\omega_2}^*(G_1,G_2) \sqsubseteq^* G_2$, nous en déduisons que les brins

et les coutures de $mcs^*_{\omega_1,\omega_2}(G_1, G_2)$ sont aussi présents dans G_1 et dans G_2. En conséquence, les deux injections sont des bijections. Donc, il existe une bijection de G_1 vers G_2 qui conserve tous les brins et toutes les coutures. Finalement de la définition 23 il suit que les nG-cartes sont isomorphes.

Preuve de la propriété 3 Pour des raisons de lisibilité, nous noterons $m_{ij} = mcs^*_{\omega_1,\omega_2}(G_i, G_j)$, $taille_{\omega_1,\omega_2}(G) = taille(G)$ et $S_{ij} = max(taille(G_i), taille(G_j))$, et nous montrons la propriété 3 en considérant séparément les deux cas suivants :

(Cas 1) : $d^{*,\omega_1,\omega_2}_{mcs}(G_1, G_2) + d^{*,\omega_1,\omega_2}_{mcs}(G_2, G_3) \geq 1$.
Dans ce cas, l'inégalité triangulaire est montrée de façon triviale car $d^{*,\omega_1,\omega_2}_{mcs}(G_1, G_3) \leq 1$.

(Cas 2) : $d^{*,\omega_1,\omega_2}_{mcs}(G_1, G_2) + d^{*,\omega_1,\omega_2}_{mcs}(G_2, G_3) < 1$.
Ici, montrons tout d'abord qu'il existe au moins une partie commune de G_2 qui appartient à m_{12} et m_{23}, *i.e.*,

$$taille(G_2) < taille(m_{12}) + taille(m_{23}) \tag{4.1}$$

Cette inéquation peut être prouvée en considérant toutes les relations d'ordre possibles entre les tailles des 3 nG-cartes.

Cas(a) : Si $taille(G_1) \geq taille(G_3) \geq taille(G_2)$, alors :
(Cas 2) $\Leftrightarrow 1 - \frac{taille(m_{12})}{taille(G_1)} + 1 - \frac{taille(m_{23})}{taille(G_3)} < 1$ (avec définition 33, et aussi $S_{12} = taille(G_1)$ et $S_{23} = taille(G_3)$)
$\Leftrightarrow taille(G_3) < \frac{taille(G_3)}{taille(G_1)} taille(m_{12}) + taille(m_{23})$ (en multipliant par $taille(G_3)$)
$\Rightarrow taille(G_3) < taille(m_{12}) + taille(m_{23})$ (car $\frac{taille(G_3)}{taille(G_1)} \leq 1$)
$\Rightarrow taille(G_2) < taille(m_{12}) + taille(m_{23})$ (car $taille(G_3) \geq taille(G_2)$).

Cas(b) : Si $taille(G_1) \geq taille(G_2) \geq taille(G_3)$, alors :
(Cas 2) $\Leftrightarrow 1 - \frac{taille(m_{12})}{taille(G_1)} + 1 - \frac{taille(m_{23})}{taille(G_2)} < 1$ (avec définition 33, et aussi $S_{12} = taille(G_1)$ et $S_{23} = taille(G_2)$)
$\Leftrightarrow taille(G_2) < \frac{taille(G_2)}{taille(G_1)} taille(m_{12}) + taille(m_{23})$ (en multipliant par $taille(G_2)$)
$\Rightarrow taille(G_2) < taille(m_{12}) + taille(m_{23})$ (car $\frac{taille(G_2)}{taille(G_1)} < 1$).

Cas(c) : Si $taille(G_2) \geq taille(G_1) \geq taille(G_3)$, alors :
(Cas 2) $\Leftrightarrow 1 - \frac{taille(m_{12})}{taille(G_2)} + 1 - \frac{taille(m_{23})}{taille(G_2)} < 1$ (avec définition 33, et aussi $S_{12} = taille(G_2)$ et $S_{23} = taille(G_2)$)
$\Leftrightarrow taille(G_2) < taille(m_{12}) + taille(m_{23})$ (en multipliant par $taille(G_2)$)

Les trois autres cas sont déduits de la propriété 2 car $d^{*,\omega_1,\omega_2}_{mcs}(G_1, G_3) = d^{*,\omega_1,\omega_2}_{mcs}(G_3, G_1)$. Ineq. (4.1) montre que la somme des tailles de deux sous-cartes communes m_{12} et m_{23} est toujours strictement supérieure à la taille de G_2. Il y a donc au moins une partie commune dénotée G_c qui appartient à la fois à m_{12} et m_{23}. On veut maintenant montrer que cette partie commune est plus petite que la plus grande sous-carte commune entre m_{12} et m_{23} (i.e. m_{123}), et que cette dernière est plus petite que m_{13}.

$$taille(m_{13}) \geq taille(m_{123}) \geq taille(G_c) > 0 \tag{4.2}$$

$taille(m_{13}) \geq taille(m_{123})$ est déduit trivialement de la définition de sous-carte définition 24. Nous savons (4.1) que $0 < taille(m_{12}) + taille(m_{23}) - taille(G_2)$.

Notons $N_o = \omega_1 \cdot B_o + \omega_2 \cdot C_o$ la taille de la partie de la nG-carte de G_2 n'étant ni dans m_{12} ni dans m_{23}, avec B_o étant le nombre de brins et C_o le nombre de coutures.

Notez que l'ensemble des brins et coutures de la partie de la nG-carte de G_2 n'étant ni dans m_{12} ni dans m_{23} n'est pas obligatoirement une nG-carte. C'est pourquoi nous ne considérons que sa taille N_o
Alors
$taille(G_2) = taille(m_{12}) + taille(m_{23}) - taille(G_c) + N_o$ donc $taille(G_c) = taille(m_{12}) + taille(m_{23}) - taille(G_2) + N_o$
Sachant que $N_o \geq 0$ nous en déduisons :

$$taille(G_c) \geq taille(m_{12}) + taille(m_{23}) - taille(G_2) \tag{4.3}$$

Or, G_c est commune à m_{12} et m_{23}, donc $taille(m_{m_{12},m_{23}}) \geq tailleG_c$.
Enfin en combinant le fait que $taille(m_{m_{12},m_{23}}) \geq tailleG_c$, $taille(m_{13}) \geq taille(m_{123})$ et que $0 < taille(m_{12}) + taille(m_{23}) - taille(G_2)$ on obtient (4.2).

En conséquence, la nG-carte $mcs^*_{\omega_1,\omega_2}(m_{12}, m_{23})$ est une sous-carte commune de G_1, G_2, et G_3 dont la taille est supérieure ou égale à $taille(m_{12}) + taille(m_{23}) - taille(G_2)$. Cette nG-carte donne une borne minimale sur la taille de la plus grande sous-carte commune de G_1 et G_3, *i.e.*,

$$taille(m_{13}) \geq taille(m_{12}) + taille(m_{23}) - taille(G_2) \tag{4.4}$$

Utilisons cette borne pour montrer que l'inégalité triangulaire est vérifiée. En développant la propriété 3 conformément à la définition 33, nous obtenons :

$$\begin{aligned} & d^{*,\omega_1,\omega_2}_{mcs}(G_1, G_3) \leq d^{*,\omega_1,\omega_2}_{mcs}(G_1, G_2) + d^{*,\omega_1,\omega_2}_{mcs}(G_2, G_3) \\ \Leftrightarrow\ & 1 - \frac{taille(m_{13})}{S_{13}} \leq 1 - \frac{taille(m_{12})}{S_{12}} + 1 - \frac{taille(m_{23})}{S_{23}} \\ \Leftrightarrow\ & taille(m_{13}) \geq \frac{S_{13}}{S_{12}} taille(m_{12}) + \frac{S_{13}}{S_{23}} taille(m_{23}) - S_{13} \end{aligned} \tag{4.5}$$

Montrons (4.5) en considérant toutes les relations d'ordre entre les tailles de nG-cartes :

(Cas 2.1) : $taille(G_1) \geq taille(G_2) \geq taille(G_3)$ on a $S_{13} = taille(G_1), S_{12} = taille(G_1), S_{23} = taille(G_2)$. Ineq. (4.5) devient $taille(m_{13}) \geq taille(m_{12}) + \frac{taille(G_1)}{taille(G_2)} taille(m_{23}) - taille(G_1)$. Comme $taille(m_{13}) \geq taille(m_{12}) + taille(m_{23}) - taille(G_2)$ (Ineq. (4.4)), nous devons montrer que $taille(m_{23}) - taille(G_2) \geq \frac{taille(G_1)}{taille(G_2)} taille(m_{23}) - taille(G_1)$, *i.e.*, $taille(m_{23}) \leq taille(G_2)$ (car $taille(G_2) - taille(G_1) < 0$). C'est une conséquence directe de la définition 33.

(Cas 2.2) : $taille(G_2) \geq taille(G_1) \geq taille(G_3)$ donc $S_{13} = taille(G_1), S_{12} = taille(G_2), S_{23} = taille(G_2)$. Ineq. (4.5) devient $taille(m_{13}) \geq \frac{taille(G_1)}{taille(G_2)} taille(m_{12}) + \frac{taille(G_1)}{taille(G_2)} taille(m_{23}) - taille(G_1)$. Comme $\frac{taille(G_1)}{taille(G_2)} \leq 1$, Ineq. (4.4) implique que $taille(m_{13}) \geq \frac{taille(G_1)}{taille(G_2)}(taille(m_{12}) + taille(m_{23}) - taille(G_2)$. En conséquence, Ineq. (4.5) est vérifiée.

(Cas 2.3) : $taille(G_1) \geq taille(G_3) \geq taille(G_2)$ on a donc $S_{13} = taille(G_1), S_{12} = taille(G_1), S_{23} = taille(G_3)$. Ineq. (4.5) devient $taille(m_{13}) \geq taille(m_{12}) + \frac{taille(G_1)}{taille(G_3)} taille(m_{23}) - taille(G_1)$. Comme $taille(m_{13}) \geq taille(m_{12}) + taille(m_{23}) - taille(G_2)$ (Ineq. (4.4)), nous devons montrer que $taille(m_{23}) - taille(G_2) \geq \frac{taille(G_1)}{taille(G_3)} taille(m_{23}) - taille(G_1)$, *i.e.*, $taille(m_{23}) \leq taille(G_3)\frac{taille(G_2)-taille(G_1)}{taille(G_3)-taille(G_1)}$ (car $taille(G_3) - taille(G_1) < 0$). Cette inégalité est vérifiée en utilisant définition 32 car $\frac{taille(G_2)-taille(G_1)}{taille(G_3)-taille(G_1)} \geq 1$.

Les trois autres cas peuvent être obtenus en inversant G_1 et G_3 car $d_{mcs}^{*,\omega_1,\omega_2}(G_1,G_3) = d_{mcs}^{*,\omega_1,\omega_2}(G_3,G_1)$. □

4.3 Distance d'édition entre nG-cartes

Nous avons présenté dans la Section 2.3, la distance d'édition entre graphes. Cette distance permet de comparer les graphes en évaluant le coût des modifications (ajouts, suppressions et substitutions de sommets et d'arêtes) nécessaires pour modifier un graphe en un autre : plus le coût est faible plus les graphes sont proches.

Opérations d'édition

Nous définissons ici un équivalent de cette distance dans le cas des nG-cartes. Nous considérons dans un premier temps des cartes généralisées non étiquetées, pour lesquelles les seules opérations d'édition considérées sont l'insertion et la suppression de brins et de coutures. Nous définirons ensuite, en Section 4.5, la notion de carte étiquetée, et nous étendrons notre distance d'édition à ces nG-cartes en introduisant des opérations de substitution. Pour cela, il faut tout d'abord définir l'ensemble des modifications considérées pour modifier une nG-carte en une autre. Ces modifications seront des opérations d'édition.

Définition 34 (Opération d'édition)
Une opération d'édition δ est une opération modifiant une nG-carte G en une nG-carte $\delta(G)$.

Nous proposons quatre opérations, deux étant basées sur les brins, et les deux autres sur les coutures. Nous utilisons par la suite uniquement des opérations menant toujours à des nG-cartes "valides", c'est-à-dire respectant les contraintes (2) et (3) de la définition 16. En particulier, une couture (b,i,b') ne peut être ajoutée (resp. supprimée) sans ajouter (resp. supprimer) la couture (b',i,b).

L'opération del_E supprime un ensemble de brins E et i-découd tous les brins non supprimés qui étaient i-cousus avec un brin supprimé.

Opération (del_E)
Soient $G=(B,\alpha_0,\ldots,\alpha_n)$ une nG-carte et $E \subseteq B$ un sous ensemble de brins.
$del_E(G) = (B',\alpha'_0,\ldots,\alpha'_n)$ *avec* $B' = B \setminus E$ *et* $\forall b' \in B', \forall i \in [0,n]$ *:*
- *si* $\alpha_i(b') \in B'$, *alors* $\alpha'_i(b') = \alpha_i(b')$ *;*
- *sinon* $\alpha'_i(b') = b'$.

L'opération $add_{E,F}$ est l'opération inverse. Elle ajoute un nouvel ensemble de brins E et un nouvel ensemble de coutures F. Les coutures ajoutées peuvent coudre à la fois des brins de E entre eux, mais aussi des brins de E avec d'autres brins de G.

Opération ($add_{E,F}$)
Soient $G=(B,\alpha_0,\ldots,\alpha_n)$ une nG-carte, E un ensemble de brins isolés tel que $E \cap B = \emptyset$ et F un ensemble de coutures tel que $\forall (b_1,i,b_2) \in F$, $\{b_1,b_2\} \cap E \neq \emptyset$ et $\alpha_i(b_1) = b_1$ et $\alpha_i(b_2) = b_2$.
$add_{E,F}(G) = (B',\alpha'_0,\ldots,\alpha'_n)$ *avec* $B' = B \cup E$ *et* $\forall b' \in B'$, $\forall i \in [0,n]$ *:*
- *si* $\exists (b',i,b'') \in F$ *:* $\alpha'_i(b') = b''$ *;*
- *sinon si* $b' \in E$ *:* $\alpha'_i(b') = b'$ *;*
- *sinon* $\alpha'_i(b') = \alpha_i(b')$.

L'opération sew_F ajoute un nouvel ensemble de coutures F.

Opération (sew_F)
Soient $G = (B, \alpha_0, \ldots, \alpha_n)$ *une* n*G-carte et* F *un ensemble de coutures tel que* $\forall (b_1, i, b_2) \in F, \alpha_i(b_1) = b_1$ *et* $\alpha_i(b_2) = b_2$. $sew_F(G) = (B, \alpha'_0, \ldots, \alpha'_n)$ *avec* $\forall i \in [0, n], \forall b' \in B$ *:*
- *si* $\exists (b', i, b'') \in F$ *alors* $\alpha'_i(b') = b''$ *;*
- *sinon* $\alpha'_i(b') = \alpha_i(b')$.

Enfin l'opération $unsew_F$ supprime un ensemble de coutures F.

Opération ($unsew_F$)
Soient $G = (B, \alpha_0, \ldots, \alpha_n)$ *une* n*G-carte et* $F \subseteq coutures_G(B)$ *un ensemble de coutures.* $unsew_F(G) = (B, \alpha'_0, \ldots, \alpha'_n)$ *avec* $\forall i \in [0, n], \forall b' \in B$ *:*
- *si* $\exists (b', i, b'') \in F$ *alors* $\alpha'_i(b') = b'$ *;*
- *sinon* $\alpha'_i(b') = \alpha_i(b')$.

La Fig. 4.6 illustre les quatre opérations.

Chemin d'édition et distance d'édition

Définissons maintenant un chemin d'édition comme une séquence de ces opérations d'édition qui modifie une nG-carte en une autre nG-carte.

Définition 35 (Chemin d'édition)
Soit $\Delta =< \delta_1, \ldots, \delta_k >$ *une séquence de* k *opérations d'édition.* Δ *est un chemin d'édition pour* G *si* $\delta_k(\delta_{k-1}(\ldots(\delta_1(G))))$*, noté* $\Delta(G)$*, est une* n*G-carte.*

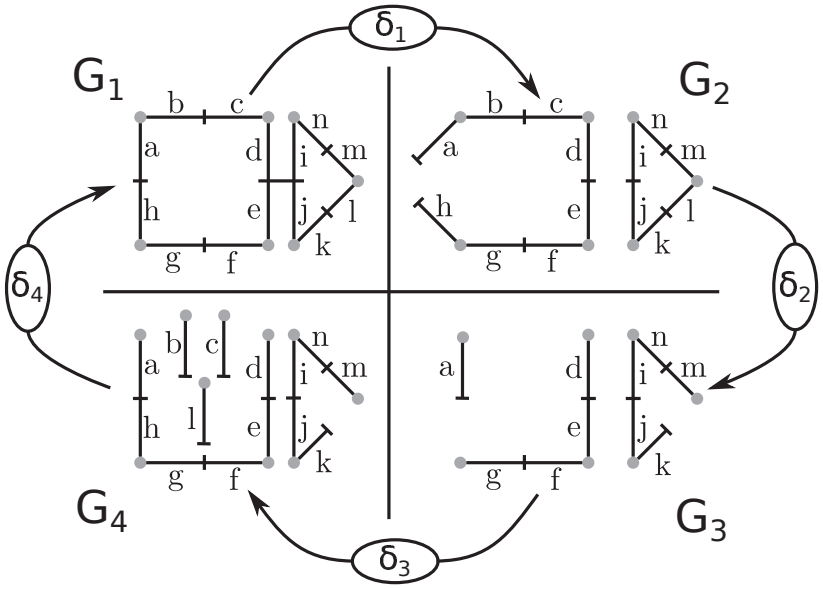
Soit Δ un chemin d'édition et δ une opération d'édition. Nous utiliserons l'opérateur $\in$ pour dénoter l'appartenance de δ à Δ, i.e., $\delta \in \Delta$ ssi δ est une opération d'édition du chemin Δ.

Notez que la validité d'un chemin dépend de la nG-carte sur laquelle il est appliqué. Par exemple, l'opération de suppression du brin h dans la 2G-carte G_1 de l'exemple Fig. 4.7 n'est pas valide, au sens où elle ne transforme pas G_1 en une 2G-carte valide. Ainsi, les opérations et chemins d'éditions sont toujours associés à une nG-carte et doivent toujours mener à une nG-carte qui respecte les contraintes de validité (2) et (3) de la définition 16.

L'opération de concaténation des chemins d'édition permet de rassembler les opérations de deux chemins au sein d'un même chemin.

Définition 36 (Concaténation de chemins d'édition)
Soient $\Delta_1 =< \delta^1_1, \ldots, \delta^1_k >$ *un chemin d'édition sur* G *et* $\Delta_2 =< \delta^2_1, \ldots, \delta^2_l >$ *un chemin d'édition sur* $\Delta_1(G)$*. La concaténation des deux chemins* Δ_1 *et* Δ_2 *est définie par* $\Delta_1 \cdot \Delta_2 =< \delta^1_1, \ldots, \delta^1_k, \delta^2_1, \ldots, \delta^2_l >$ *et modifie la* n*G-carte* G *en la* n*G-carte* $\Delta_1.\Delta_2(G) = \Delta_2(\Delta_1(G))$.

Remarquez que $\Delta_1 \cdot \Delta_2(G)$ est un chemin d'édition car $\Delta_1 \cdot \Delta_2(G)$ est une nG-carte. La Fig. 4.7 illustre les chemins d'édition. Dans cet exemple, la concaténation des deux chemins d'édition $\Delta_{G_1 \to G_2} \cdot \Delta_{G_2 \to G_3}$ permet de modifier G_1 en G_3 en passant par G_2. $\Delta_{G_1 \to G_3}$ décrit un second chemin d'édition pour transformer la nG-carte G_1 en G_3 sans passer par la 2G-carte G_2. Notez que si l'on considère le nombre d'opérations réalisées, le chemin $\Delta_{G_1 \to G_3}$ est plus court que le chemin $\Delta_{G_1 \to G_2} \cdot \Delta_{G_1 \to G_2}$.



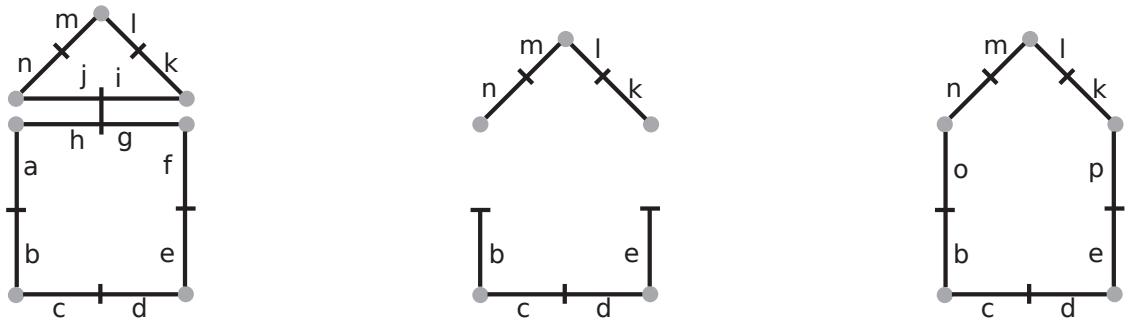
$$\begin{aligned} G_2 &= \delta_1(G_1) \text{ avec } \delta_1 = \{unsew_{F_1}\} \text{ et} \\ F_1 &= \{(a,0,h),(h,0,a),(d,2,i),(i,2,d),(e,2,j),(j,2,e)\} \end{aligned}$$

$$\begin{aligned} G_3 &= \delta_2(G_2) \text{ avec } \delta_2 = \{del_{E_2}\} \text{ et} \\ E_2 &= \{b,c,l,h\} \end{aligned}$$

$$\begin{aligned} G_4 &= \delta_3(G_3) \text{ avec } \delta_3 = \{add_{E_3,F_3}\} \text{ et} \\ E_3 &= \{b,c,l,h\}, F_3 = \{(h,0,a),(a,0,h),(h,1,g),(g,1,h)\} \end{aligned}$$

$$\begin{aligned} G_1 &= \delta_4(G_1) \text{ avec } \delta_4 = \{sew_{F_4}\} \text{ et} \\ F_4 &= \{(c,0,b),(b,0,c),(k,0,l),(l,0,k),(a,1,b),(b,1,a),(c,1,d),(d,1,c), \\ &\quad (m,1,l),(l,1,m),(d,2,i),(i,2,d),(e,2,j),(j,2,e)\} \end{aligned}$$

FIGURE 4.6 – *Exemples d'opérations $\delta_1, \delta_2, \delta_3, \delta_4$,appliquées successivement sur une 2G-carte. δ_1 découd les brins a et h ainsi que les deux faces, δ_2 supprime les brins $\{b,c,l,h\}$, δ_3 ajoute des brins ainsi que certaines involutions, et enfin δ_4 recoud les involutions restantes.*



(a) G_1 (b) G_2 (c) G_3

$$\Delta_{G_1 \to G_2} =< del_{\{a,f,g,h,i,j\}} >$$

$$\Delta_{G_2 \to G_3} =< add_{\{o,p\}\{(b,0,o),(o,0,b),(e,0,p),(p,0,e),(n,1,o),(o,1,n),(k,1,p),(p,1,k)\}} >$$

$$\Delta_{G_1 \to G_3} =< del_{\{g,h,i,j\}}, sew_{\{(a,1,n),(n,1,a),(f,1,k),(k,1,f)\}} >$$

FIGURE 4.7 – *$\Delta_{G_1 \to G_2}$, $\Delta_{G_2 \to G_3}$ et $\Delta_{G_1 \to G_3}$ sont trois chemins d'édition. Le premier nommé $\Delta_{G_1 \to G_2}$ explicite les opérations permettant de transformer la 2G-carte G_1 en G_2, en supprimant les brins $\{a, f, g, h, i, j\}$ ainsi que la totalité des coutures associées. Le chemin $\Delta_{G_2 \to G_3}$ donne les opérations permettant de modifier la 2G-carte G_2 pour obtenir G_3 en ajoutant les brins o et p. Enfin $\Delta_{G_1 \to G_3}$ décrit une autre solution pour transformer G_1 en une carte isomorphe à G_3 sans passer par G_2.*

La distance d'édition est définie comme le coût du plus court chemin d'édition existant entre deux nG-cartes. Le coût de ce chemin est calculé en utilisant une fonction établissant un coût pour chaque opération.

Définition 37 (Distance d'édition entre nG-cartes)
Soit c une fonction qui associe à chaque opération δ un coût $c(\delta) \in \mathbb{R}^+$. Nous notons $c(\Delta)$ la somme des coûts des opérations impliquées par le chemin d'édition Δ, i.e. $c(\Delta) = \sum_{\delta_i \in \Delta}(c(\delta_i))$.
La distance d'édition entre G et G' est définie par $d^c_{ed}(G, G') = \min_{\Delta \in \mathcal{D}}(c(\Delta))$, où $\mathcal{D}$ est l'ensemble des chemins d'édition transformant G et G'.

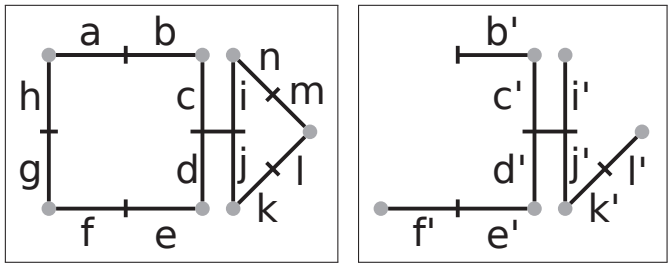
Considérons par exemple les 2G-cartes de la Fig. 4.7, et la fonction de coût telle que le coût d'ajout ou de suppression d'un brin ou d'une couture soit de 1 :

- $c(del_E) = |E| + |\{(b_1, i, b_2) \in coutures_G(B), b_1 \in E \text{ ou } b_2 \in E\}|$. Le coût associé à la suppression de brins est donc égal au nombre de brins et de coutures supprimés ;
- $c(add_{E,F}) = |E| + |F|$. Le coût associé à l'ajout de brins est égal au nombre de brins et de coutures ajoutées ;
- $c(sew_F) = c(unsew_F) = |F|$. Le coût associé aux coutures et découtures est égal au nombre de coutures ajoutées/supprimées.

Alors, dans la Fig. 4.7, $d^c_{ed}(G_1, G_2)$ est égal à 26 et correspond aux 6 brins supprimés et aux 20 coutures impactées, $d^c_{ed}(G_2, G_3)$ est égal à 10 et correspond aux 2 brins et aux 8 coutures ajoutés. Enfin, $d^c_{ed}(G_1, G_3)$ est égal à 24 et correspond aux 4 brins supprimés et leurs 16 coutures, ainsi que les 4 coutures ajoutées.

4.4 Relation entre d_{mcs} et d_{ed}

Dans le cas des graphes, Bunke met en évidence dans [Bun97] un lien entre la distance d'édition de graphes et le plus grand sous-graphe commun. En effet, il montre



(a) G (b) G'

b	b'	c'	d'	e'	f'	i'	j'	k'	l'
f(b)	b	c	d	e	f	i	j	k	l

(c) fonction de sous-isomorphisme induit $G' \rightarrow G$

$$i\Delta_f^{G \rightarrow G'} =< del_{\{a,g,h,m,n\}} >$$

$$i\Delta_f^{G' \rightarrow G} =< add_{\{a',g',h',m',n'\},F} >$$

$$\text{Avec } F = \begin{array}{l} \{(g',0,h'),(h',0,g'),(n',0,m'),(m',0,n'),(b',0,a'),(a',0,b'),(g',1,f') \\ (f',1,g'),(h',1,a'),(a',1,h'),(n',1,i'),(i',1,n'),(m',1,l'),(l',1,m')\} \end{array}$$

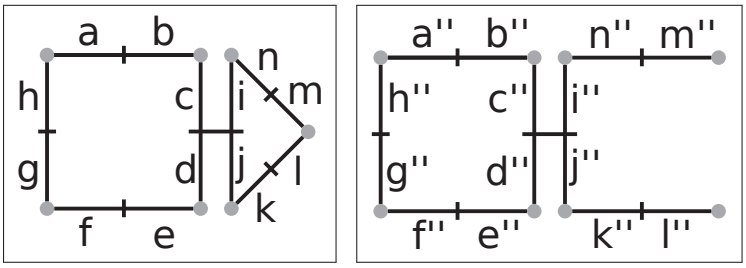
FIGURE 4.8 – *Exemple de sous-carte induite G' et de sa fonction de sous-isomorphisme induite f. $i\Delta_f^{G \rightarrow G'}$ représente la suite d'opérations modifiant G en une 2G-carte isomorphe à G' et $i\Delta_f^{G' \rightarrow G}$ l'opération inverse modifiant G' en une 2G-carte isomorphe à G.*

que la distance d'édition est calculable (avec certains coûts) si nous connaissons le plus grand sous-graphe commun. Elle est égale au coût du chemin d'édition modifiant le premier graphe vers le plus grand sous-graphe commun puis modifiant le plus grand sous-graphe commun vers le second graphe. Nous allons ici mettre en évidence un lien similaire entre la distance d'édition de nG-cartes et la plus grande sous-carte commune. Tout d'abord, nous proposons de construire un chemin d'édition à partir d'une sous-carte commune. Puis nous montrons que le coût de ce chemin d'édition est minimum et par conséquent que son coût est égale à la distance d'édition.

Définition 38 (Chemin d'édition associé à une fonction de sous-isomorphisme induit) *Soient G et G' deux nG-cartes telles que $G \sqsubseteq^i G'$, et soit $f : B \rightarrow B'$ la fonction de sous-isomorphisme associée. Sans perte de généralité, nous supposons $B \cap B' = \emptyset$. Soit E l'ensemble des brins qui ne sont pas appariés par f, i.e., $E = \{b' \in B' | \nexists b \in B, f(b) = b'\}$ et soit F l'ensemble des coutures entre les brins ajoutés et les brins de G, i.e., $F = \{(b'_1, i, b'_2) \in coutures_{G'}(B') | b'_1 \in E$ ou $b'_2 \in E\}$. Nous définissons les chemins d'édition associés à la fonction de sous-isomorphisme induit comme suit :*

- $i\Delta_f^{G' \rightarrow G} =< del_E >$
- $i\Delta_f^{G \rightarrow G'} =< add_{E,F} >$

$i\Delta_f^{G' \rightarrow G}(G')$ supprime tous les brins qui ne sont pas appariés par f ainsi que les coutures associées. La fonction f étant la fonction de sous-isomorphisme induit, elle apparie les brins faisant partie de la sous-carte induite. Par définition de la fonction de sous-isomorphisme, toutes les coutures entre deux brins appariés sont conservées. Donc,



(a) G (b) G''

b	a”	b”	c”	d”	e”	f”	g”	h”	i”	j”	k”	l”	m”	n”
f’(b)	a	b	c	d	e	f	g	h	i	j	k	l	m	n

(c) fonction de sous-isomorphisme partiel $G'' \to G$

$$p\Delta_f^{G\to G''} =< unsew_{\{(m,1,l)(l,1,m)\}} >$$

$$p\Delta_f^{G''\to G} =< sew_{\{(m'',1,l'')(l'',1,m'')\}} >$$

FIGURE 4.9 – *Exemple de sous-carte partielle G'' et de sa fonction de sous-isomorphisme partielle f. $p\Delta_f^{G\to G''}$ représente la suite d'opérations modifiant G en une $2G$-carte isomorphe à G'' et $p\Delta_f^{G''\to G}$ l'opération inverse modifiant G'' en une $2G$-carte isomorphe à G.*

$i\Delta_f^{G'\to G}(G')$ modifie G' en une nG-carte isomorphe à G. Symétriquement, $i\Delta_f^{G\to G'}$ est l'opération inverse, et modifie la nG-carte G de telle sorte à ce que $i\Delta_f^{G\to G'}(G)$ soit une nG-carte isomorphe à G'. La Fig. 4.8 illustre les opérations déduites du sous-isomorphisme induit f.

Montrons que la concaténation des chemins d'édition associés à la fonction de sous-isomorphisme induit décrivant la plus grande sous-carte commune induite est le chemin de coût minimal pour la distance d'édition pour des coûts particuliers. Nous définissons un ensemble de coût particulier noté $c^i_{\omega_1,\omega_2}$ n'autorisant que les opérations sur les brins (add et del).

- Soient G une nG-carte et $E \subseteq B$ un ensemble de brins. $c^i_{\omega_1,\omega_2}(del_E) = c^i_{\omega_1,\omega_2}(add_{E,F}) = \omega_1.|E| + \omega_2.|\{(d_1,i,d_2) \in coutures_G(D'), d_1 \in E \text{ où } d_2 \in E\}|)$;
- Soient G une nG-carte, E un ensemble de brins tels que $B \cap E = \emptyset$ et F un ensemble de coutures sur $B \cup E$ associant un brin de B avec un brin de E. $c^i_{\omega_1,\omega_2}(sew_F) = c^i_{\omega_1,\omega_2}(unsew_F) = +\infty$.

Proposition 1
Soient $\omega_1 \in \mathbb{R}^{+*}$, G et G' deux nG-cartes, $G'' = mcs^i_{\omega_1,0}(G,G') = (B'', \alpha''_0, \ldots, \alpha''_n)$, et $f : B'' \to B$ (resp. $f' : B'' \to B'$) un isomorphisme de sous-carte induit associé à la relation de sous-isomorphisme $G'' \sqsubseteq^i G$ (resp. $G'' \sqsubseteq^i G'$).

Nous avons $d^{c^i_{1,0}}_{ed}(G,G') = \sum_{\delta \in i\Delta_f^{G\to G''} \cdot i\Delta_{f'}^{G''\to G'}} (c^i_{1,0}(\delta))$.

Démonstration. Nous noterons $c = c^i_{1,0}$ pour des raisons de lisibilité dans la suite de la preuve. Conformément à la définition 38, $i\Delta_f^{G\to G''} \cdot i\Delta_{f'}^{G''\to G'}$ est un chemin d'édition qui transforme G en G' en passant par la nG-carte G'', qui est une plus grande sous-carte commune induite de G et de G'. Pour prouver que le coût de ce chemin d'édition est égal à la distance d'édition, nous devons prouver qu'il est minimum.

Supposons qu'il existe un chemin d'édition Δ_2 tel que $c(\Delta_2) < c(i\Delta_f^{G\to G''} \cdot i\Delta_{f'}^{G''\to G'})$. Δ_2 est une combinaison de suppressions $del_{Br_0}, \ldots del_{Br_n}$ et d'insertions $add_{Ba_0,S_0}, \ldots$

add_{Ba_n,S_n} telle que $\Delta_2(G) \simeq G'$. Notez qu'un brin ne peut pas appartenir à deux opérations, sinon le brin est impliqué dans une insertion et une suppression. Cela impliquerait qu'il est possible de supprimer le brin des deux opérations et obtenir ainsi un meilleur coût. Il en suit que $\forall B_i, B'_i \in \{Br_0, \dots Br_n, Ba_0, \dots, Ba_n\}, B_i \neq B'_i \Rightarrow B_i \cap B'_i = \varnothing$. Cette observation nous permet de réorganiser les opérations de Δ_2 en commençant par les suppressions suivies des insertions : $\Delta_2 =< del_{Br_0}, \dots del_{Br_n}, add_{Ba_0,S_0}, \dots add_{Ba_n,S_n} >$. Nous pouvons alors réécrire les opérations de suppressions en une unique opération, et de même pour les opérations d'insertions tel que : $\Delta_2 =< del_{B'_r}, add_{B'_a,S'} >$, avec $B'_r = Br_0 \cup \dots \cup Br_n$, $B'_a = Ba_0 \cup \dots \cup Ba_n$ et $S' = S_0 \cup \dots \cup S_n$.

Notez que la nG-carte $del_{B'_r}(G)$ est une nG-carte composée de brins appartenant à G et aussi à G' et conservant les coutures. Par déduction $del_{B'_r}(G) \sqsubseteq^i G$ et $del_{B'_r}(G) \sqsubseteq^i G'$.

Considérons les deux parties du chemin d'édition séparément. Si nous avons $c(del_{B'_r}) < c(\Delta_f^{G \to G''})$ alors $taille_{\omega_1,0}(del_{B'_r}(G)) > taille_{\omega_1,0}(G'')$. Autrement dit il existe une nG-carte plus grande que G'' étant une sous-carte de G et de G', ce qui contredit la définition de plus grande sous-carte commune. De l'autre côté si $c(add_{Ba',S'}) < c(\Delta_{f'}^{G'' \to G'})$ nous pouvons étudier l'opération inverse de $add_{B'_a,S'}$ qui supprimerait les brins de G' pour obtenir une sous-carte commune à G et à G'. Ce qui contredit aussi la définition de la plus grande sous-carte commune car cela impliquerait que $taille_{\omega_1,0}(del_{B'_a}(G')) > taille_{\omega_1,0}(G'')$ et en conséquence il existerait une sous-carte commune de G et G' plus grande que la plus grande-sous-carte commune ce serait G''. Nous en déduisons que $c(del_{B'_r}) \geq c(i\Delta_f^{G \to G''})$ et $c(add_{Ba'_n,S'_n}) \geq c(i\Delta_{f'}^{G'' \to G'})$ alors $c(\Delta_2) \geq c(i\Delta_f^{G \to G''} \cdot i\Delta_{f'}^{G'' \to G'})$. □

La Prop. 1 permet de déduire qu'il est possible de calculer la suite d'opérations induisant le coût optimal pour la distance d'édition à partir de la plus grande sous-carte induite. Dans le cas partiel, bien que les opérations associées à la fonction de sous-isomorphisme partiel soient différentes, un résultat similaire est obtenu.

Définition 39 (Chemin d'édition associé à une fonction de sous-isomorphisme partiel) *Soient G et G' deux nG-cartes telles que $G \sqsubseteq^p G'$, et soit $f : B \to B'$ une fonction de sous-isomorphisme partiel. Sans perte de généralité, nous supposons $B \cap B' = \emptyset$. Soit F l'ensemble des coutures modifiées entre deux brins de G, i.e., $F = \{(b_1, i, b_2) \in coutures_{G'}(B') | (b_1, i, b_2) \notin coutures_{G'}(B)\}$ et soit E l'ensemble des brins isolés insérés ou supprimés, i.e., $E = \{b' \in B' | \nexists b \in B, f(b) = b'\}$.*

Nous définissons les chemins d'éditions associés à la fonction de sous-isomorphisme partiel comme suit :

- $p\Delta_f^{G' \to G} =< unsew_F, del_E >$
- $p\Delta_f^{G \to G'} =< add_{E,\emptyset}, sew_F >$

$p\Delta_f^{G' \to G}$ découd les coutures non conservées et supprime tous les brins qui ne sont pas appariés par f ainsi que les coutures associées. La fonction f étant la fonction de sous-isomorphisme partiel elle apparie les brins faisant partie de la sous-carte partielle. Tous les brins et les coutures non conservés par la fonction f sont supprimés par les opérations del et $unsew$. Donc, $p\Delta_f^{G' \to G}(G')$ est une nG-carte isomorphe à G. Symétriquement, $p\Delta_f^{G \to G'}$ est l'opération inverse, et modifie la nG-carte G en ajoutant les brins et les coutures n'existant pas dans la sous-carte de telle sorte que $p\Delta_f^{G \to G'}(G)$ soit une nG-carte isomorphe à G'. La Fig. 4.9 illustre les opérations déduites du sous-isomorphisme partiel f'.

Montrons que la concaténation des chemins d'édition associés à la fonction de sous-isomorphisme partiel décrivant la plus grande sous-carte commune partielle est le chemin de coût minimal pour la distance d'édition pour des coûts particuliers.

Pour cela nous considérons une fonction de coût particulière, notée c^p.

- Soient G une nG-carte et $E \subseteq B$ un ensemble de brins. $c^p_{\omega_1,\omega_2}(del_E) = \omega_1.|E| + \omega_2.|\{(d_1, i, d_2) \in coutures_G(B), d_1 \in E \text{ ou } d_2 \in E\}|)$;
- Soient G une nG-carte, E un ensemble de brins tels que $B \cap E = \emptyset$ et F un ensemble de coutures sur $B \cup E$ associant un brin de B avec un brin de E. $c^p_{\omega_1,\omega_2}(add_{E,F}) = \omega_1.|E| + \omega_2.|F|$;
- Soient G une nG-carte, F un ensemble de coutures sur B. $c^p_{\omega_1,\omega_2}(sew_F) = c^p_{\omega_1,\omega_2}(unsew_F) = \omega_2.|F|$.

Proposition 2
Soient ω_1, ω_2 dans $\mathbb{R}^{+*}$, $G'' = mcs^p_{\omega_1,\omega_2}(G, G') = (B'', \alpha''_0, \ldots, \alpha''_n)$, et $f : B'' \to B$ (resp. $f' : B'' \to B'$) une fonction de sous-isomorphisme partiel associée à la relation de sous-isomorphisme $G'' \sqsubseteq^p G$ (resp. $G'' \sqsubseteq^p G'$). Nous avons $d^{c^p_{1,1}}_{ed}(G, G') = \sum_{\delta_j \in p\Delta_f^{G \to G''} \cdot p\Delta_{f'}^{G'' \to G'}} (c^p_{1,1}(\delta_j))$.

Démonstration. Nous noterons $c = c^p_{1,1}$ pour des raisons de lisibilité dans la suite de la preuve. Supposons qu'il existe un chemin d'édition Δ_2 tel que $c(\Delta_2) < c(i\Delta_f^{G \to G''} \cdot i\Delta_{f'}^{G'' \to G'})$. Δ_2 est une combinaison de suppressions $del_{Br_1}, \ldots del_{Br_{k1}}$, d'insertions $add_{Ba_1,Sa_1}, \ldots add_{Ba_{k2},Sa_{k2}}$, de décousures $unsew_{Sr_1}, \ldots unsew_{Sr_{k3}}$, et de coutures $sew_{Si_1}, \ldots sew_{Si_{k4}}$ telle que $\Delta_2(G) \simeq G'$. Des coûts de c, nous en déduisons que $c(add_{E,F}) = c(add_{E,\emptyset}) + c(sew_F)$ et $c(del_F) = c(unsew_{S^F}) + c(del_F)$ où S^F est l'ensemble des coutures de G impliquant les brins de F. Le coût de l'opération d'ajout d'un brin et de ses coutures est identique aux deux opérations effectuant d'abord l'ajout du brin isolé puis de l'opération de couture. Ainsi, il est possible de dissocier les opérations d'ajouts de brins et de leurs coutures en deux opérations : une ajoutant le brin isolé et une effectuant les coutures. Symétriquement il est possible de dissocier l'opération de suppression d'un brin et des coutures associées en une opération de décousure des coutures associées puis en une opération de suppression du brin isolé. Notons $Sd_1 \ldots Sd_{k1}$ l'ensemble des coutures associées aux opérations $del_{Br_1}, \ldots del_{Br_{k1}}$. Ainsi, le chemin d'édition peut-être réécrit comme un ensemble de suppressions de brins $del_{Br_1}, \ldots del_{Br_{k1}}$, d'insertions de brins $add_{Ba_1,\emptyset}, \ldots add_{Ba_{k2},\emptyset}$, de coutures $sew_{Si_1}, \ldots sew_{Si_{k4}}, sew_{Sa_1} \ldots sew_{Sa_{k2}}$ et de décousures $unsew_{Sr_1}, \ldots unsew_{Sr_{k3}}, unsew_{Sd_1} \ldots unsew_{Sd_{k1}}$.

Notez qu'un brin ne peut pas appartenir à la fois à une opération add et à une opération del, sinon le brin est supprimé après avoir été inséré dans une opération précédente. Cela veut dire qu'il est possible de supprimer le brin des deux opérations et réduire le nombre d'opérations pour obtenir un chemin d'édition ayant un meilleur coût. De la même manière, une couture ne peut être présente dans deux opérations sew et $unsew$.

Il en suit que $\forall B_i, B'_i \in \{Br_1, \ldots Br_{k1}, Ba_1, \ldots, Ba_{k2}\}, B_i \neq B'_i \Rightarrow B_i \cap B'_i = \varnothing$ et que $\forall S_i, S'_i \in \{Sd_1, \ldots Sd_{k1}, Sr_1, \ldots, Sr_{k3}, Si_1, \ldots, Si_{k4}, Sa_1, \ldots, Sa_{k2}\}, S_i \neq S'_i \Rightarrow S_i \cap S'_i = \varnothing$.

Cette observation nous permet de réorganiser les opérations de Δ_2 en commençant par les décousures, suivies des suppressions de brins isolés, puis des insertions de brins

isolés et enfin des coutures :

$$\begin{aligned}\Delta_2 =< \quad & unsew_{Sr_1}, \dots unsew_{Sr_{k3}}, unsew_{Sd_1} \dots unsew_{Sd_{k1}}, \\ & del_{Br_1}, \dots del_{Br_{k1}}, \\ & add_{Ba_1,\emptyset}, \dots add_{Ba_{k2},\emptyset}, \\ & sew_{Si_1}, \dots sew_{Si_{k4}}, sew_{Sa_1} \dots sew_{Sa_{k2}} >\end{aligned}$$

Nous pouvons compacter chacun des types d'opérations en une opération en fusionnant les ensembles de brins et de coutures impliqués tels que

- $Sr = Sr_1 \cup \dots \cup Sr_{k3} \cup Sd_1 \cup \dots \cup Sd_{k1}$;
- $Br = Br_1 \cup \dots \cup Br_{k1}$;
- $Bi = Ba_1 \cup \dots \cup Ba_{k2}$;
- $Si = Si_1 \cup \dots \cup Si_{k4} \cup Sa_1 \cup \dots \cup Sa_{k2}$

Nous en déduisons donc que Δ_2 devient une séquence de quatre opérations :

$$\Delta_2 =< unsew_{Sr}, del_{Br}, add_{Bi,\emptyset}, sew_{Si} >$$

Notez que la nG-carte $unsew_{Sr} \cdot del_{Br}(G)$ est une nG-carte composée de brins appartenant à G et aussi à G', alors en utilisant la définition 39, nous en déduisons que $unsew_{Sr} \cdot del_{Br}(G) \sqsubseteq^p G$ et $unsew_{Sr} \cdot del_{Br}(G) \sqsubseteq^p G'$.

Considérons les deux parties du chemin d'édition séparément. Si nous avons $c(unsew_{Sr} \cdot del_{Br}(G)) < c(\Delta_f^{G \to G''})$ alors $taille_{\omega_1,\omega_2}(unsew_{Sr'} \cdot del_{Br'}(G)) > taille_{\omega_1,\omega_2}(G'')$, ce qui signifie qu'il existe une nG-carte commune à G et à G' plus grande que la plus grande sous-carte commune de G et G'. Il y a donc une contradiction.

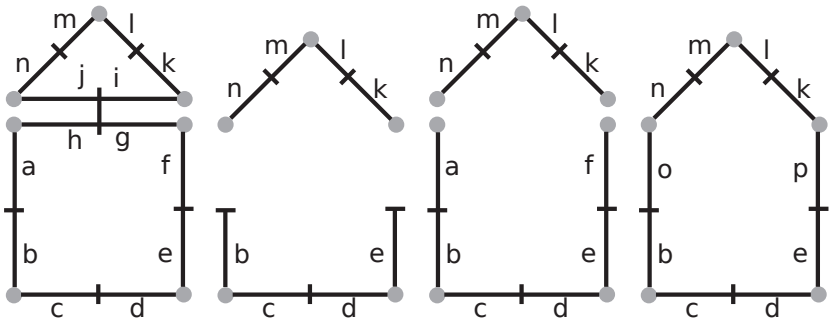
De l'autre côté si $c(add_{Bi,\emptyset} \cdot sew_{Si}) < c(\Delta_{f'}^{G'' \to G'})$ nous pouvons étudier l'opération inverse de $add_{Bi,\emptyset} \cdot sew_{Si}$ notée $delBi \cdot sew_{Si}$ qui découdrait et supprimerait les brins de G' pour obtenir une sous-carte commune à G et à G'.

Cela contredit aussi la définition de la plus grande sous-carte commune car cela impliquerait que $taille_{\omega_1,\omega_2}(unsew_{Sr} \cdot del_{Br}(G)) > taille_{\omega_1,\omega_2}(G'')$. Encore une fois, nous obtenons une contradiction car une sous-carte commune de G et G' est plus grande que la plus grande sous-carte commune de G et G'. Nous en déduisons que $c(unsew_{Sr} \cdot del_{Br}) \geq c(i\Delta_f^{G \to G''})$ et $c(add_{Bi,\emptyset} \cdot sew_{Si}) \geq c(i\Delta_{f'}^{G'' \to G'})$ alors $c(\Delta_2) \geq c(i\Delta_f^{G \to G''} \cdot i\Delta_{f'}^{G'' \to G'})$. □

Il est donc possible de calculer la distance d'édition entre deux nG-cartes G et G' en calculant la plus grande sous-carte commune $mcs^*_{\omega_1,\omega_2}(G, G')$. En effet, pour des coûts particuliers, le chemin d'édition de coût minimum est obtenu en transformant G en $mcs^*_{\omega_1,\omega_2}(G, G')$ puis en tranformant $mcs^*_{\omega_1,\omega_2}(G, G')$ en G'.

La figure 4.10 illustre cela. En considérant la distance d'édition uniquement avec les opérations de modifications basées sur les brins, nous obtenons une distance de $d_{ed}^{c^i_{\omega_1,\omega_2}}(G, G') = 8.\omega_1 + 28.\omega_2$. Notez que la pondération permet de modifier l'importance des brins par rapport aux involutions. Dans ce cas, étant donné que nous ne considérons que les opérations sur les brins, pondérer le coefficient ω_2 avec une valeur de 0 est parfaitement réaliste et la distance ne tiendra alors compte que des modifications de brins. En considérant toutes les opérations (avec les opérations de coutures et décousures), nous trouvons une distance de $d_{ed}^{c^p_{\omega_1,\omega_2}}(G, G') = 4.\omega_1 + 20.\omega_2$, ce qui exprime une plus forte proximité des deux nG-cartes.

Un autre exemple est donné dans la Fig. 4.11. L'unique modification réalisée est la décousure des deux faces. Dans le cas où nous ne considèrons que les brins pour les opérations, nous obtenons une distance de $4.\omega_1 + 16.\omega_2$, qui reste très élevée par rapport au



(a) G (b) $mcs^{i}_{1,1}(G,G')$ (c) $mcs^{p}_{1,1}(G,G')$ (d) G'

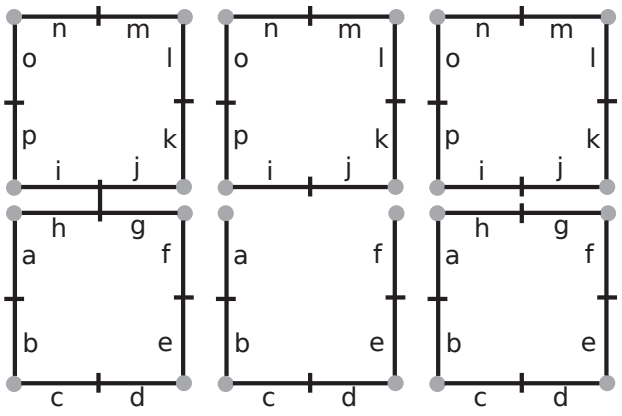
$$\Delta_{G \to mcs^{i}_{1,1}(G,G')} =< del_{\{a,h,g,f,i,j\}} >$$

$$\Delta_{mcs^{i}_{1,1}(G,G') \to G'} =< add_{\{o,p\}\{(b,0,o),(o,0,b),(e,0,p),(p,0,e),(n,1,o),(o,1,n),(k,1,p),(p,1,k)\}} >$$

$$\Delta_{G \to mcs^{p}_{1,1}(G,G')} =< del_{\{i,j,g,h\}} >$$

$$\Delta_{mcs^{p}_{1,1}(G,G') \to G'} =< sew_{\{(a,1,n),(n,1,a),(f,1,k),(k,1,f)\}} >$$

FIGURE 4.10 – *La distance d'édition $d_{ed}^{c^{i}_{\omega_1,\omega_2}}(G,G')$ est égale au coût du chemin d'édition $\Delta_{G \to mcs^{i}_{1,1}(G,G')} \cdot \Delta_{mcs^{i}_{1,1}(G,G') \to G'}$. Elle est égale à $8.\omega_1 + 28.\omega_2$, avec $d_{ed}^{c^{i}_{\omega_1,\omega_2}}(G, mcs^{i}_{1,1}(G,G')) = 6.\omega_1 + 20.\omega_2$ et $d_{ed}^{c^{i}_{\omega_1,\omega_2}}(mcs^{i}_{1,1}(G,G'),G') = 2.\omega_1 + 8.\omega_2$. La distance d'édition $d_{ed}^{c^{p}_{\omega_1,\omega_2}}(G,G')$ est égale au coût du chemin d'édition $\Delta_{G \to mcs^{p}_{1,1}(G,G')} \cdot \Delta_{mcs^{p}_{1,1}(G,G') \to G'}$. Elle est égale à $4.\omega_1 + 20\omega_2$ avec $d_{ed}^{c^{p}_{\omega_1,\omega_2}}(G, mcs^{p}_{1,1}(G,G')) = 4.\omega_1 + 16.\omega_2$ et $d_{ed}^{c^{p}_{\omega_1,\omega_2}}(mcs^{p}_{1,1}(G,G'),G') = 4.\omega_2$.*



(a) G (b) $mcs^{i}_{1,0}(G,G')$ (c) G'

$$\Delta_{G\rightarrow mcs^{i}_{1,1}(G,G')} =< del_{\{g,h\}} >$$

$$\Delta_{mcs^{i}_{1,1}(G,G')\rightarrow G'} =< add_{\{g,h\}\{(g,0,h),(h,0,g),(a,1,h),(h,1,a),(g,1,f),(f,1,g)\}} >$$

$$\Delta_{G\rightarrow mcs^{p}_{1,1}(G,G')} =< unsew_{\{(i,2,h),(h,2,i),(j,2,g),(g,2,j)\}} >$$

FIGURE 4.11 – *La distance d'édition $d^{c^{i}_{\omega_1,\omega_2}}_{ed}(G,G')$ est égale au coût du chemin d'édition $\Delta_{G\rightarrow mcs^{i}_{1,1}(G,G')}\cdot\Delta_{mcs^{i}_{1,1}(G,G')\rightarrow G'}$. Elle est égale à $4.\omega_1 + 16.\omega_2$ avec $d^{c^{i}_{\omega_1,\omega_2}}_{ed}(G, mcs^{i}_{1,0}(G,G')) = 2.\omega_1 + 10.\omega_2$ et $d^{c^{i}_{\omega_1,\omega_2}}_{ed}(mcs^{i}_{1,0}(G,G'),G') = 2.\omega_1 + 6.\omega_2$. La distance d'édition $d^{c^{p}_{\omega_1,\omega_2}}_{ed}(G,G')$ est égale au coût du chemin d'édition $\Delta_{G\rightarrow G'}$ car $G' \simeq mcs^{p}_{\omega_1,\omega_2}(G,G')$. Elle est égale à $4.\omega_2$.*

nombre de modifications effectivement réalisées pour passer d'une 2G-carte à l'autre. Si nous considérons les opérations liées aux coutures cette fois, la distance est beaucoup plus proche $d^{c^{p}_{\omega_1,\omega_2}}_{ed}(G,G') = 4.\omega_2$. Cette seconde distance réagit donc mieux aux opérations de décousures.

Dans la Section 6 nous illustrerons plus en détail cette distance d'édition pour comparer des images et des maillages.

4.5 nG-cartes étiquetées et distance d'édition

Les distances présentées dans les sections 4.2 et 4.3 permettent de quantifier la différence structurelle entre deux nG-cartes. Elles ne tiennent compte que des brins et des coutures. Nous avons aussi vu dans le cas des graphes que de l'information peut être portée par les sommets ou arêtes sous la forme d'étiquettes pour aider à la description de l'objet modélisé. Il est également possible dans le cas des nG-cartes d'ajouter de l'information aux brins. Pour cela, nous ajoutons à chaque brin une étiquette en étendant la définition de nG-carte généralisée.

Définition 40 (Carte généralisée étiquetée)
Soit $n \geq 1$. Une nG-carte généralisée étiquetée, est un tuple $G = (B, E, e, \alpha_0, \ldots, \alpha_n)$ où :

- *$(B, \alpha_0, \ldots, \alpha_n)$ est une nG-carte ;*
- *E est un ensemble d'étiquettes ;*
- *$e : B \to E \cup \{\epsilon\}$, est une fonction qui associe une étiquette à chaque brin de la nG-carte.*

ϵ est l'étiquette utilisée pour indiquer qu'un brin est non etiqueté.

Il faut alors ajouter une opération à la distance d'édition présentée en Section 4.3 pour prendre en compte en plus de l'information structurelle, une information portée par des étiquettes sur les brins. La nouvelle distance décrira ainsi en plus de la similarité entre leurs structures, la similarité entre les informations portées par les brins.

Ainsi nous définissons une opération de substitution pour permettre la substitution de brins possédant une étiquette, avec d'autres brins possédant une étiquette différente.

L'opération $subs_{D,l}$ substitue les étiquettes des brins de D par d'autres étiquettes suivant la fonction l.

Opération ($subs_{D,l}$)
Soient $G = (B, E, e, \alpha_0, \ldots, \alpha_n)$ une nG-carte étiquetée, $D \subseteq B$ un ensemble de brins, et l une fonction $l : D \to E$ associant une étiquette $l(b)$ à chaque brin $b \in D$.
$subs_{D,l}(G) = (B, E, e', \alpha_0, \ldots, \alpha_n)$ *avec* $\forall b \in B$ *:*

- $e'(b) = l(b)$ *, si* $b \in D$ *;*
- $e'(b) = e(b)$ *, sinon*

Les opérations d'ajout, et de suppression de brins étant définies pour les nG-cartes non étiquetées nous définissons une version modifiée des opérations d'ajout et de suppression de brins utilisant les étiquettes.

Opération (del'_D)
Soient $G = (B, E, e, \alpha_0, \ldots, \alpha_n)$ une nG-carte étiquetée et $D \subseteq B$ un sous ensemble de brins tel que $\forall b \in D, e(b) = \epsilon$
$del_D(G) = (B', E, e', \alpha'_0, \ldots, \alpha'_n)$ *avec* $\forall b \in B', e'(b) = e(b)$ *et* $\forall i \in [0, n]$ *:*

- *si* $\alpha_i(b') \in B \setminus D$, *alors* $\alpha'_i(b') = \alpha_i(b')$ *;*
- *sinon* $\alpha'_i(b') = b'$.

Opération ($add'_{D,F}$)
Soit $G = (B, E, e, \alpha_0, \ldots, \alpha_n)$ une nG-carte étiquetée, D un ensemble de brins isolés tel que $D \cap B = \emptyset$ et F un ensemble de coutures tel que $\forall (b_1, i, b_2) \in F$, $\{b_1, b_2\} \cap D \neq \emptyset$ et $\alpha_i(b_1) = b_1$, $\alpha_i(b_2) = b_2$:
$add_{D,F}(G) = (B', E, e', \alpha'_0, \ldots, \alpha'_n)$ *avec* $B' = B \cup D$ *et* $\forall b \in B, e'(b) = e(b)$ *et* $\forall b \in D, e'(b) = \epsilon$ *et* $\forall b' \in B', \forall i \in [0, n]$ *:*

- *si* $\exists (b', i, b'') \in F : \alpha'_i(b') = b''$ *;*
- *sinon si* $b' \in D : \alpha'_i(b') = b'$ *;*
- *sinon* $\alpha'_i(b') = \alpha_i(b')$.

Avec ces nouvelles opérations seuls des brins étiquetés avec ϵ peuvent être ajoutés ou supprimés. L'opération de substitution permet de modifier l'étiquette du brin en ϵ avant une suppression, et de modifier l'étiquette ϵ en une autre étiquette après une insertion. Il est donc possible de définir un coût de substitution de l'étiquette ϵ vers une

autre étiquette et inversement, correspondant au coût d'insertion (ou de suppression) d'un brin en fonction de son étiquette.

Les étiquettes représentant différentes informations en fonction des objets modélisés, il conviendra de choisir une fonction de coût pour les opérations d'édition dépendant de son type d'information. Dans le Chap. 6, nous proposerons différentes étiquettes, adaptées à la comparaison d'images et de maillages basées sur les couleurs, les aires de faces et les angles.

Distance d'édition étiquetée

L'opération de substitution ayant été définie, la distance d'édition est directement étendue en définissant un coût pour l'opération de substitution.

Notons $cost(l_1, l_2)$ le coût de substitution de l'étiquette l_1 par l'étiquette l_2. Nous adaptons les fonctions de coûts $c^i_{\omega_1,\omega_2}$ et $c^p_{\omega_1,\omega_2}$ pour intégrer les étiquettes. Soit $c^{e*}_{\omega_1,\omega_2}$ avec $* \in \{i, p\}$une fonction de coût intégrant les étiquettes telle que :

- $c^{e*}_{\omega_1,\omega_2}(add'_{D,F}) = c^*_{\omega_1,\omega_2}(add_{D,F})$;
- $c^{e*}_{\omega_1,\omega_2}(del'_D) = c^*_{\omega_1,\omega_2}(delD)$;
- $c^{e*}_{\omega_1,\omega_2}(subs_{D,l}) = \sum_{b\in D}(cost(e(b), l(b)))$.

Notez que nous considérons les opérations del' et add' pour c^e au lieu des opérations del et add.

Nous présenterons un exemple de cette distance dans le Chap. 6.

4.6 Discussion

Dans ce chapitre nous avons tout d'abord défini la notion de sous-isomorphisme partiel, plus tolérant aux erreurs que la notion de sous-isomorphisme induit. En effet, le sous-isomorphisme induit, contrairement au sous-isomorphisme partiel, n'identifie pas en tant que sous-carte une nG-carte pour laquelle aurait seulement été décousus un certain nombre de brins. Après avoir proposé une taille paramétrable, nous avons proposé une distance basée sur la taille de la plus grande partie commune entre deux nG-cartes. Puis, nous avons proposé une vision plus opérationnelle, en étendant la distance d'édition de graphes au domaine des nG-cartes. Pour cela nous avons défini un ensemble d'opérations permettant la modification de nG-cartes. Cette distance permet de mesurer la similarité entre nG-cartes en évaluant le coût des modifications nécessaires pour transformer une nG-carte à partir d'une autre. Notez que ces méthodes sont uniquement basées sur la structure des nG-cartes représentant les objets. C'est pourquoi nous avons ajouté une opération de substitution décrivant une opération de changement d'étiquette sur les brins.

Chapitre 5

Algorithmes d'approximation de distances entre nG-cartes

Sommaire

5.1 Appariements et sous-cartes 51
5.2 Calcul d'une approximation de la plus grande sous-carte commune . 58
5.3 Heuristiques pour choisir le prochain couple de brins à apparier . . 61
5.4 Évaluation expérimentale de ApproxD 68
5.5 Discussion 75

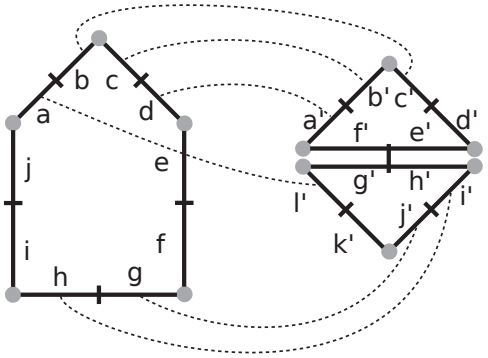
Dans ce chapitre nous proposons un algorithme permettant de calculer une approximation supérieure (i.e., une valeur supérieure ou égale) des distances présentées dans le Chap. 4. Nous allons tout d'abord définir la notion d'appariement permettant de mettre en relation les brins de deux nG-cartes comparées. Puis nous montrons que nous pouvons déduire une sous-carte commune de cet appariement. Nous proposons un algorithme glouton construisant un appariement et nous introduisons des heuristiques (Section 5.3) visant à construire un appariement correspondant à une sous-carte commune s'approchant de la plus grande sous-carte commune. Nous en déduisons une approximation supérieure de la distance d_{mcs} ainsi qu'une approximation de la distance d'édition qui découle du chemin décrit par la sous-carte commune calculée. Enfin, nous évaluons notre algorithme sur un jeu de nG-cartes générées aléatoirement.

5.1 Appariements et sous-cartes

Pour calculer la distance d_{mcs} entre deux nG-cartes G et G', nous avons besoin de calculer la plus grande sous-carte commune $mcs(G,G') = G''$ caractérisée par deux relations de sous-isomorphisme $f_{G''\to G}$ et $f_{G''\to G'}$. La composition de ces deux fonctions $f_{G''\to G'} \circ f^{-1}_{G''\to G}$ apparie des brins de G avec des brins de G' (ceux qui sont retrouvés dans G'').

Nous appelons *appariement* cette fonction de mise en correspondance de brins de G avec des brins de G'. Notons que cet appariement est univoque, i.e., chaque brin de G (resp. G') est apparié à au plus un brin de G' (resp. G).

Définition 41 (Appariement univoque)
Un appariement univoque de G et G' est une relation $m \subseteq B \times B'$ telle que $\forall (b_i, b'_i), (b_j, b'_j) \in m^2, b_i = b_j \Leftrightarrow b'_i = b'_j$.



$$m = \{(a,l'),(b,c'),(c,b'),(d,a'),(g,j'),(h,i')\}$$
$$B_{\downarrow m} = \{a,b,c,d,g,h\}$$
$$B'_{\downarrow m} = \{a',b',c',i',j',l'\}$$

FIGURE 5.1 – *Exemple d'appariement m entre la 2G-carte G et la 2G-carte G'. L'appariement est représenté ici en pointillés.*

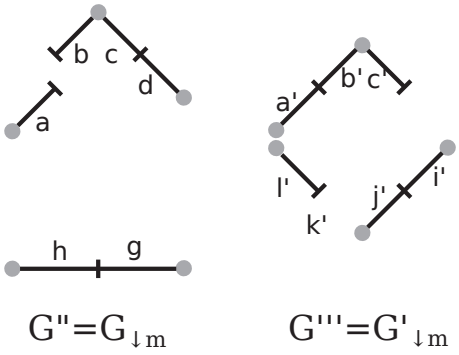


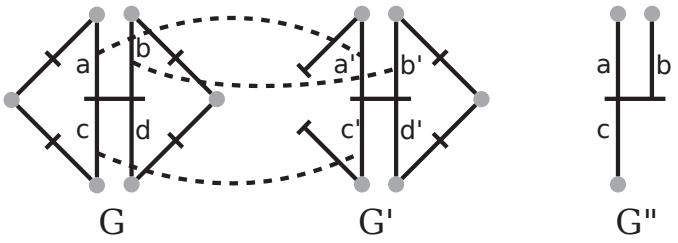
FIGURE 5.2 – *G'' est la sous-carte extraite de G par l'appariement m, et G''' est la sous-carte extraite de G' par l'appariement m de la Fig. 5.1.*

Nous noterons $m(b)$ le brin apparié à b par m, i.e., si $(b,b') \in m$ alors $m(b) = b'$ et $m(b') = b$. Sinon, $m(b)$ sera égal à ϵ si b n'est pas apparié, i.e., si $\forall b' \in B', (b,b') \notin m$ alors $m(b) = \epsilon$. Nous noterons $B_{\downarrow m}$ (resp. $B'_{\downarrow m}$) les brins de B (resp. B') appariés par m, i.e. $B_{\downarrow m} = \{b \in B | m(b) \neq \epsilon\}$.

La Fig. 5.1 donne un exemple d'appariement entre deux nG-cartes. Nous définissons la sous-carte extraite d'un appariement, composée des brins appariés et des coutures conservées.

Définition 42 (Sous-carte extraite d'un appariement)
Soient G et G' deux nG-cartes. La sous-carte extraite d'un appariement univoque $m \subseteq B \times B'$ est $G_{\downarrow m} = (B_{\downarrow m}, \alpha''_0, \ldots, \alpha''_n)$ telle que $\forall b \in B_{\downarrow m}, \forall i \in [0,n]$:
- *si $\alpha_i(b) \in B_{\downarrow m}$ et $\alpha'_i(m(b)) = m(\alpha_i(b))$ alors $\alpha''_i(b) = \alpha_i(b)$;*
- *sinon $\alpha''_i(b) = b$.*

Dans la Fig. 5.2 nous illustrons les deux 2G-cartes G'' et G''' extraites par l'appariement m de la Fig. 5.1. Notez que $G_{\downarrow m}$ est isomorphe à $G'_{\downarrow m}$. Cependant, tous les appariements ne permettent pas d'extraire une sous-carte valide. Par exemple dans la Fig. 5.3, l'ensemble de brins et coutures G'' extraits de l'appariement m ne respecte pas la contrainte d'intégrité imposant que $\alpha_0 \circ \alpha_2$ soit une involution. Il convient donc



brin	a	b	c	d
m(brin)	a'	b'	c'	ϵ

FIGURE 5.3 – *Exemple d'un appariement non valide. m est un appariement entre la nG-carte G et la nG-carte G'. G'' n'est pas une nG-carte. Elle ne respecte pas la contrainte d'intégrité $\alpha_0 \circ \alpha_2$ est une involution.*

d'ajouter des contraintes sur les appariements pour garantir qu'une sous-carte extraite est valide. Nous définissons donc l'*appariement induit valide* qui permet l'extraction d'une sous-carte induite valide, et l'*appariement partiel valide* qui permet l'extraction d'une sous-carte partielle valide. Notez qu'un appariement induit valide est par définition un appariement partiel valide c'est pourquoi nous définissons tout d'abord l'appariement partiel valide.

Définition 43 (Appariement partiel valide)
Soient G et G' deux nG-cartes. Un appariement univoque $m \subseteq B \times B'$ est partiel valide si $\forall b \in B_{\downarrow m}, \forall i,j \in [0,n]$ tel que $i+2 \leq j$:
$[\alpha_i(b) \in B_{\downarrow m} \wedge m(\alpha_i(b)) = \alpha'_i(m(b)) \wedge \alpha_j(b) \in B_{\downarrow m} \wedge m(\alpha_j(b)) = \alpha'_j(m(b))]$
$\Leftrightarrow m(\alpha_i(\alpha_j(b))) = \alpha'_i(\alpha'_j(m(b)))$.

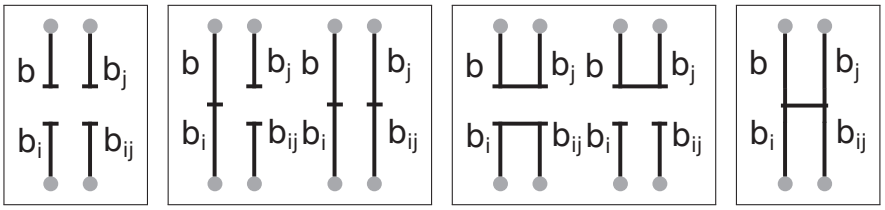
En d'autres termes, si un brin b est apparié à un brin b' et que ses voisins par α_i et α_j sont aussi appariés aux voisins de b' par α'_i et α'_j, alors le brin $\alpha_i(\alpha_j(b))$ doit être apparié au brin $\alpha'_i(\alpha'_j(b'))$. Cette propriété garantit que l'appariement permet d'extraire une sous-carte valide. Dans la Fig. 5.3, l'appariement n'est pas valide car les brins a, b et c sont appariés aux brins a', b' et c' alors que le brin d n'est pas apparié au brin d'. En conséquence, $\alpha_0 \circ \alpha_2$ n'est pas une involution dans la sous-carte extraite. La contrainte de la définition 43 interdit ce cas.

Ces contraintes de validité sur l'appariement permettent d'extraire de façon unique et simple une nG-carte valide à partir d'un appariement. Autrement dit, si m est un appariement valide entre deux nG-cartes G et G' alors la sous-carte décrite par m est composée des brins appariés, et des coutures préservées par l'appariement.

Théorème 2
Soient G et G' deux nG-cartes et m un appariement partiel valide entre G et G', la nG-carte $G'' = G_{\downarrow m}$ extraite de l'appariement m est valide.

Démonstration. Selon la définition 16, dans une nG-carte si $\forall i \in [0,n], \alpha_i$ est une involution et si $\forall (i,j) \in [0,n]^2$ avec $i+2 \leq j$, $\alpha_i \circ \alpha_j$ est une involution. Soient $i,j \in [0,n]^2$ tel que $i+2 \leq j$ et $b \in B$. Nous étudions les différents cas possibles. Plaçons-nous tout d'abord dans le cas où les coutures entre deux brins appariés sont conservées, i.e., si $(\{b, \alpha_i(b)\} \in B_{\downarrow m}$ alors $\alpha'_i(m(b)) = m(\alpha_i(b)))$ et de manière identique pour j, $(\{b, \alpha_j(b)\} \in B_{\downarrow m}$ alors $\alpha'_j(m(b)) = m(\alpha_j(b)))$.

(Cas 1) : b est i-libre et j-libre : $\alpha_i(b) = b$ et $\alpha_j(b) = b$
Alors $\alpha''_i(b) = b$ et $\alpha''_j(b) = b$ donc $\alpha''_i \circ \alpha''_j$ est une involution



(a) Cas 1 (b) Cas 2 et 5 (c) Cas 3 et 5 (d) Cas 4 et 6

FIGURE 5.4 – *Cas de la preuve, nous avons choisi pour l'illustration* $i = 0$ *et* $j = 2$.

(Cas 2) : b est i-cousu et j-libre : $\alpha_i(b) = b_i \neq b$ et $\alpha_j(b) = b$
Comme G est une nG-carte, $\alpha_i \circ \alpha_j$ est une involution, cela implique que $\alpha_j(b_i) = b_i$. Nous distinguons donc tous les cas d'appariement possibles des brins b et b_i

(Cas 2.1) : b et b_i ne sont pas appariés, i.e., $m(b) = \epsilon$ et $m(b_i) = \epsilon$
Aucun des deux brins n'est conservé dans G'', cas résolu trivialement (il n'y a aucun des brins dans G'').

(Cas 2.2) : b est apparié et b_i n'est pas apparié, i.e., $m(b) \neq \epsilon$ et $m(b_i) = \epsilon$
Alors $\alpha''_i(b) = b$ et $\alpha''_j(b) = b$ donc $\alpha''_i \circ \alpha''_j$ est une involution.

(Cas 2.3) : b n'est pas apparié et b_i est apparié, i.e., $m(b) = \epsilon$ et $m(b_i) \neq \epsilon$
Alors $\alpha''_i(b_i) = b_i$ et $\alpha''_j(b_i) = b_i$ donc $\alpha''_i \circ \alpha''_j$ est une involution.

(Cas 2.4) : b et b_i sont appariés, i.e., $m(b) \neq \epsilon$ et $m(b_i) \neq \epsilon$
Nous avons $\alpha_j(b) = b$ et $\alpha_j(b_i) = b_i$, donc $\alpha''_j(b) = b$, $\alpha''_j(b_i) = b_i$ et $\alpha''_i(b) = b_i$ donc $\alpha''_i \circ \alpha''_j$ est une involution.

(Cas 3) : b est i-libre et j-cousu : peut être déduit du Cas 2, par symétrie de i et j.

(Cas 4) : b est i-cousu et j-cousu : $\alpha_i(b) = b_i \neq b$ et $\alpha_j(b) = b_j \neq b$
G est une nG-carte, de la définition 16 il suit que $\alpha_i \circ \alpha_j$ est une involution. Donc $\alpha_j(\alpha_i(b)) = \alpha_i(\alpha_j(b)) = b_{ij}$ avec $b_{ij} \notin \{b, b_i, b_j\}$.
Nous étudions tous les cas possible d'appariement.

(Cas 4.1) : Aucun brin n'est apparié, i.e., $m(b) = \epsilon, m(b_i) = \epsilon, m(b_j) = \epsilon$ et $m(b_{ij}) = \epsilon$
Aucun des brins n'est conservé dans la sous-carte extraite, cas résolu trivialement (il n'y a aucun des brins dans G'').

(Cas 4.2) : Un seul brin est apparié, i.e, $m(b) = \epsilon$ ou $m(b_i) = \epsilon$ ou $m(b_j) = \epsilon$ ou $m(b_{ij}) = \epsilon$
Un seul des brins b, b_i, b_j et b_{ij} est apparié, seul ce brin est conservé dans la sous-carte extraite. Nommons b' ce brin apparié nous avons $\alpha''_i(b') = b'$ $\alpha''_j(b') = b'$ donc $\alpha''_i \circ \alpha''_j$ est une involution.

(Cas 4.3) : Deux brins de $\{b, b_i, b_j b_{ij}\}$ sont appariés, c'est-à-dire $b', b'' \in \{b, b_i, b_j b_{ij}\}, b' \neq b'', m(b') \neq \epsilon$ et $m(b'') \neq \epsilon$

(Cas 4.3a) : b' et b'' ne sont pas en relation par une involution.
Nous avons $\alpha_i(b') \neq b''$ et $\alpha_j(b') \neq b''$, or ce sont les deux seuls brins appariés. Nous en déduisons que nous avons $m(\alpha_i(b')) = \epsilon$, $m(\alpha_j(b')) = \epsilon$, $m(\alpha_i(b'')) = \epsilon$, $m(\alpha_j(b')) = \epsilon$, donc $\alpha''_i(b') = b'$, $\alpha''_j(b') = b'$, $\alpha''_i(b'') = b''$, $\alpha''_j(b'')) = b''$.

(Cas 4.3b) : $\alpha_i(b') = b''$.
b' et b'' sont les deux seuls brins appariés, donc $m(\alpha_j(b')) = \epsilon$ et $m(\alpha_j(b')) = \epsilon$. Nous en déduisons que $\alpha''_i(b') = b''$, $\alpha''_j(b') = b'$ et $\alpha''_j(b'') = b''$, donc $\alpha''_i \circ \alpha''_j$ est une involution.

(Cas 4.3c) : $\alpha_j(b') = b''$.
Cas résolu par symétrie de i et j a partir du Cas 4.3b.

(Cas 4.4) : b, b_i, b_j, b_{ij} sont tous appariés.
Nous avons $\alpha_i(b) = b_i, \alpha_j(b) = b_j, \alpha_i(\alpha_j(b)) = b_{ij}$, or $m(b) \neq \epsilon$, $m(b_i) \neq \epsilon$, $m(b_j) \neq \epsilon$, $m(b_{ij}) \neq \epsilon$, donc comme toutes les coutures sont conservées $\alpha''_i(b) = b_i, \alpha''_j(b) = b_j$, $\alpha''_j(b_i) = b_{ij}$ et $\alpha''_i(b_j) = b_{ij}$. $\alpha''_i \circ \alpha''_j$ est une involution.

(Cas 4.5) : Seulement trois brins de b, b_i, b_j, b_{ij} sont appariés.
Ce cas mène toujours à un appariement invalide.

(Cas 4.5a) : b, b_i et b_j sont appariés, i.e., $m(b) \neq \epsilon$, $m(b_i) \neq \epsilon$, $m(b_j) \neq \epsilon$ et $m(b_{ij}) = \epsilon$. Selon la définition 43, si m est valide nous devons avoir $m(b_{ij}) \neq \epsilon$. Nous avons donc une contradiction avec l'hypothèse de départ m est un appariement valide.

(Cas 4.5b) : Cas similaires pour tout les différents car de 3 brins appariés sur les 4.

Maintenant, plaçons nous dans le cas où toutes les coutures entre les brins ne sont pas nécessairement conservées, c'est-à-dire que deux brins étant reliés par une involution appariés ne conserve pas forcément la couture, i.e., pour deux brins b et b_i tel que $b_i = \alpha_i(b)$ et $m(b) \neq \epsilon$ et $m(b_i) \neq \epsilon$ nous avons $\alpha'_i(m(b)) \neq m(\alpha_i(b))$. Alors $\alpha''_i(b) = b$ et $\alpha''_i(b_i) \neq b_i$. Nous étudions les cas où deux, trois ou quatre brins sont appariés, le cas avec un unique brin apparié ne permettant pas de conserver de coutures.

(Cas 5a) : b est i-libre mais j-cousu.
Si la j-couture n'est pas conservée $\alpha'_j(m(b)) \neq m(\alpha_j(b))$ alors $\alpha''_j(b) = b$ et $\alpha''_j(\alpha_j(b)) = \alpha_j(b)$. Comme $\alpha''_i(b) = b$ alors $\alpha''_i \circ \alpha''_j$ est une involution.

(Cas 5b) : b est i-cousu mais j-libre.
Cas résolu par symétrie de i et j.

(Cas 6) : b est i-cousu et j-cousu.
$\alpha_i(b) = b_i \neq b$ et $\alpha_j(b) = b_j \neq b$
G est une nG-carte, de la définition 16 il suit que $\alpha_i \circ \alpha_j$ est une involution.
Donc $\alpha_j(\alpha_i(b)) = \alpha_i(\alpha_j(b)) = b_{ij}$ avec $b_{ij} \notin \{b, b_i, b_j\}$.
Nous étudions tous les cas d'appariement.

(Cas 6.1) : Deux brins de $\{b, b_i, b_j, b_{ij}\}$ sont appariés, soient $b', b'' \in \{b, b_i, b_j, b_{ij}\}, b' \neq b'', m(b') \neq \epsilon$ et $m(b'') \neq \epsilon$
Si la couture est préservée, nous sommes dans le Cas 3, si elle ne l'est pas, nous sommes dans le Cas 5, $\alpha''_i \circ \alpha''_j$ est une involution.

(Cas 6.2) : Trois brins de $\{b, b_i, b_j, b_{ij}\}$ sont appariés.
Si toutes les coutures sont préservées, alors m est invalide (Cas 3.5). Si une unique couture n'est pas préservée, alors nous avons deux brins dans le Cas 2, et un brin dans le Cas 1. Si deux coutures ne sont pas préservées, alors les trois brins appariés sont dans le Cas 1.

(Cas 6.3) : Quatre brins sont appariés, $m(b) \neq \epsilon$, $m(b_i) \neq \epsilon$, $m(b_j) \neq \epsilon$, $m(b_{ij}) \neq \epsilon$

(Cas 6.3a) : Aucune couture n'est préservée.
Si aucune couture n'est préservée, les quatre brins sont dans le Cas 1. Autrement dit $m(b), m(b_i), m(b_j), m(b_{ij}$ sont i-libres et j-libres donc $\alpha''_i \circ \alpha''_j$ est une involution.

(Cas 6.3b) : Une unique couture est préservée.
Si une unique couture est préservée, deux brins sont dans le Cas 2, et deux brins dans le Cas 1.

(Cas 6.3c) : Uniquement deux coutures sont préservées.
Si seulement deux coutures sont préservées et de même dimension, nous avons deux couples de brins dans le Cas 2. Si les deux coutures sont de dimensions différentes, alors nous avons $\alpha_i(b) \neq b$ et $\alpha_j(b) \neq b$. Comme seulement deux coutures sont préservées, nous n'avons pas $m(\alpha_i(\alpha_j(b))) = \alpha'_i(\alpha'_j(b))$ ou $m(\alpha_j(\alpha_i(b))) = \alpha'_j(\alpha'_i(b))$, donc d'après la définition 43, m n'est pas valide. Nous avons donc une contradiction, ce cas n'est pas possible.

(Cas 6.3d) : Trois coutures sont préservées.
Si seulement trois coutures sont préservées, alors $\exists b' \in \{b, b_i, b_j, b_{ij}\}$ tel que nous avons $\alpha_i(b') \neq b'$ et $\alpha_j(b') \neq b'$. Comme seulement trois coutures sont préservées, nous n'avons pas $m(\alpha_i(\alpha_j(b))) = \alpha'_i(\alpha'_j(b))$ ou $m(\alpha_j(\alpha_i(b))) = \alpha'_j(\alpha'_i(b))$, donc d'après la définition 43, m n'est pas valide. Nous avons donc une contradiction, ce cas n'est pas possible.

□

Montrons que la sous-carte extraite d'un appariement m valide à partir de G est isomorphe à la sous-carte extraite de l'appariement m à partir de G'. Cette propriété est importante afin que le sens de l'appariement n'influence pas le résultat de la sous-carte extraite.

Théorème 3
Soient G et G' deux nG-cartes et m un appariement partiel valide entre G et G', les nG-cartes $G_{\downarrow m}$ et $G'_{\downarrow m}$ extraites de l'appariement m sont isomorphes.

Démonstration. Nous avons $G_{\downarrow m} = (B_{\downarrow m}, \alpha''_0, \ldots, \alpha''_n)$ et $G'_{\downarrow m} = (B'_{\downarrow m}, \alpha'''_0, \ldots, \alpha'''_n)$. m définit une bijection entre $B_{\downarrow m}$ et $B'_{\downarrow m}$. De la définition de sous-carte extraite (définition 42), nous savons que $(b_1, i, b_2) \in coutures_{G_{\downarrow m}}(B_{\downarrow m})$ si $\alpha'_i(m(b_1)) = m(\alpha_i(b_1))$ avec $\alpha_i(b_1) = b_2$, nous savons aussi que $(b'_1, i, b'_2) \in coutures_{G'_{\downarrow m}}(B'_{\downarrow m})$ si $\alpha'_i(m(b'_1)) = m(\alpha_i(b'_1))$ avec $\alpha_i(b'_1) = b'_2$. m étant une bijection, nous avons donc conservation de tous les brins et de toutes les coutures de $G_{\downarrow m}$ et $G'_{\downarrow m}$ ce qui est la définition de l'isomorphisme. □

Dans le cas induit, une contrainte est ajoutée pour garantir que toutes les coutures entre deux brins appariés sont conservées.

Algorithme 1: EXTRACT(G, G', m)

Données : Deux nG-cartes $G = (B, \alpha_0, \ldots, \alpha_n)$, $G' = (B', \alpha'_0, \ldots, \alpha'_n)$, un appariement univoque m valide de G vers G'
Résultat : Retourne $G'' = (B'', \alpha''_0, \ldots, \alpha''_n)$, la carte extraite de l'appariement m.

```
1 B'' ← B_{↓m}
2 Pour tout b ∈ B_{↓m} faire
3     Pour tout i ∈ [0, n] faire
4         Si α_i(b) ∈ B_{↓m} et α'_i(m(b)) = m(α_i(b)) alors
5             α''_i(b) ← m(α_i(b))
6         Sinon
7             α''_i(b) ← b
8 retourner G''
```

Définition 44 (Appariement induit valide)
Soient G et G' deux nG-cartes. Un appariement univoque $m \subseteq B \times B'$ est induit valide s'il est partiel valide et s'il préserve toutes les involutions entre les brins appariés i.e.*,* $\forall b \in B_{\downarrow m}, \forall i \in [0, n]$ *:*

- *si b est i-libre et $m(b)$ est i-cousu alors $\alpha'_i(m(b)) \notin B'_{\downarrow m}$;*
- *si b est i-cousu et $m(b)$ est i-libre alors $\alpha_i(b) \notin B_{\downarrow m}$;*
- *si les deux brins b et $m(b)$ sont i-cousus alors $m(\alpha_i(b)) = \alpha'_i(m(b))$ ou ($\alpha_i(b) \notin B'_{\downarrow m}$ et $\alpha'_i(m(b)) \notin B'_{\downarrow m}$).*

Un appariement partiel valide est un appariement induit valide si toutes les coutures existantes entre deux brins appariés sont conservées dans la seconde nG-carte. Par exemple, dans la Fig. 5.1, l'appariement est partiel valide, mais n'est pas induit valide car la couture $(a, 0, b)$ n'est pas conservée alors que les brins a et b sont appariés, i.e., $(m(a), 0, m(b)) = (c', 0, l') \notin coutures_{G'}(B')$. Les théorèmes 2 et 3 sont facilement étendus au cas induit.

De la définition 42, découle directement un algorithme d'extraction de sous-carte à partir d'un appariement valide en temps linéaire.

Cet algorithme (Algorithme 1) initialise tout d'abord B'' avec les brins de B qui sont appariés. Il parcourt ensuite les brins et les dimensions pour recopier les coutures de G et G' conservées par l'appariement m dans G''. Notons que l'algorithme est identique que la sous-carte extraite soit une sous-carte induite ou une sous-carte partielle. En effet, le type de la sous-carte extrait dépend uniquement de l'appariement, les contraintes de validité de l'appariement induit et partiel permettent de décider si la sous-carte est induite ou partielle.

Nous avons défini un appariement permettant de décrire la sous-carte commune de deux nG-cartes. Nous allons proposer un algorithme calculant cet appariement afin que sa sous-carte extraite soit le plus proche possible de la plus grande sous-carte commune.

Algorithme 2: APPROXD$(G, G', *)$

Données : Deux nG-cartes $G = B, \alpha_0, \ldots, \alpha_n)$, $G' = (B', \alpha'_0, \ldots, \alpha'_n)$, $(\omega_1, \omega_2) \in R^{2+}$ et $* \in \{i, p\}$

Résultat : Retourne une borne supérieure de $d^{*,\omega_1,\omega_2}_{mcs}(G, G')$

1 $m \leftarrow \emptyset$
2 *Cand*$\leftarrow B \times B'$
3 **Tant que** Cand$\neq \emptyset$ **faire**
4 **Répéter**
5 $(b, b') \leftarrow$ Choisir(*Cand*)
6 Retirer (b, b') de *Cand*
7 $m' \leftarrow$ Propager(m, b, b')
8 **jusqu'à** $estValide^*(m')$ **ou** *Cand*$= \emptyset$;
9 **Si** $estValide^*(m')$ **alors**
10 $m \leftarrow m'$
11 Enlever de *Cand* tout couple (b_i, b'_i) tel que $m(b_i) \neq \epsilon$ ou $m(b'_i) \neq \epsilon$.
12 $G'' \leftarrow Extract(G, G', m)$
13 **retourner** $1 - \frac{taille_{\omega_1,\omega_2}(G'')}{max(taille_{\omega_1,\omega_2}(G), taille_{\omega_1,\omega_2}(G'))}$

5.2 Calcul d'une approximation de la plus grande sous-carte commune

Le problème de recherche d'une plus grande sous-carte commune est un problème NP-Difficile. Dans la mesure où les nG-cartes décrivant des images ont généralement plusieurs milliers de brins, nous avons délibérément choisi de ne pas développer un algorithme exact calculant la plus grande sous-carte commune. En effet, un tel algorithme aurait très probablement été inutilisable pour des nG-cartes de taille réaliste. Nous avons donc choisi de développer un algorithme glouton calculant une solution approchée en utilisant une heuristique. Un algorithme glouton construit une solution en réalisant des choix sans jamais les remettre en cause : partant d'un appariement vide, notre algorithme construit un appariement en appariant des couples de brins qui ne seront jamais désappariés. L'objectif est de construire un appariement qui s'approche au plus de la plus grande sous-carte commune, donc qui apparie un maximum de brins tout en conservant un maximum de coutures.

L'algorithme ApproxD (Algorithme 2) construit un appariement à partir de deux nG-cartes et retourne une approximation de la distance $d^{*,\omega_1,\omega_2}_{mcs}$. La ligne 1 initialise l'appariement m à vide. La ligne 2 initialise *Cand* avec l'ensemble de tous les couples de brins pouvant être potentiellement appariés. Ensuite l'algorithme entre dans une boucle pour essayer successivement d'apparier les couples de brins de *Cand*. La boucle $4-8$ sélectionne un couple de brins dans *Cand* en suivant une heuristique donnée (décrite dans la Section 5.3). La fonction $Propager$ ajoute ce couple à m et effectue aussi une propagation de l'appariement à d'autres couples de brins afin de tenter de résoudre les éventuels problèmes de validité d'appariements. Si l'appariement obtenu est valide, alors m et *Cand* sont mis à jour.

L'algorithme $Propager$ (Algorithme 3) apparie deux brins et tente de résoudre les problèmes de validité de l'appariement. Pour cela, nous étudions le voisinage du couple de brins à apparier et vérifions (Ligne 5) que ce dernier n'entraîne pas un appariement

Algorithme 3: PROPAGER(m, b, b')

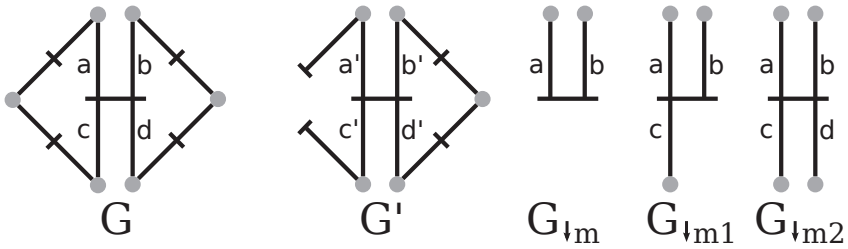
Données : Un appariement valide m, et deux brins non appariés b et b'
Résultat : Un appariement appariant en plus des brins appariés dans m, les brins b et b' ainsi que les couples propagés pour essayer de satisfaire la validité de l'appariement

1 **Si** $m(b) \neq \epsilon$ ***ou*** $m(b') \neq \epsilon$ **alors**
2 **retourner** m
3 Ajouter (b, b') dans m.
4 **Pour tout** $i \in [0, n]$ ***et*** $j \in [i+2, n]$ **faire**
5 **Si** $\alpha_i \circ \alpha_j(b)$ *n'est pas une involution dans* $Extract(G, G', m)$ **alors**
6 **Si** $m(\alpha_i(b)) = \epsilon$ **alors**
7 $m \leftarrow Propager(m, \alpha_i(b), \alpha'_i(b'))$
8 **Sinon Si** $m(\alpha_j(b)) = \epsilon$ **alors**
9 $m \leftarrow Propager(m, \alpha_j(b), \alpha'_j(b'))$
10 **Sinon**
11 $m \leftarrow Propager(m, \alpha_i(\alpha_j(b)), \alpha'_i(\alpha'_j(b')))$
12 **retourner** m

invalide. Il est possible de résoudre localement les problèmes d'intégrité en ajoutant un autre couple de brins. Pour apparier ce nouveau couple, nous appelons récusivement la fonction $Propager$ (Ligne 6-11) pour résoudre d'éventuelles violations de la contrainte d'intégrité par ces nouveaux couples. Dans l'exemple, Fig. 5.5, l'appariement m est valide mais apparier le brin c avec le brin c' l'invalide. C'est la fonction $Propager$ qui, pour satisfaire la contrainte de validité de la 2G-carte extraite de l'appariement, apparie en plus le brin d avec le brin d', ce qui rend l'appariement de nouveau valide.

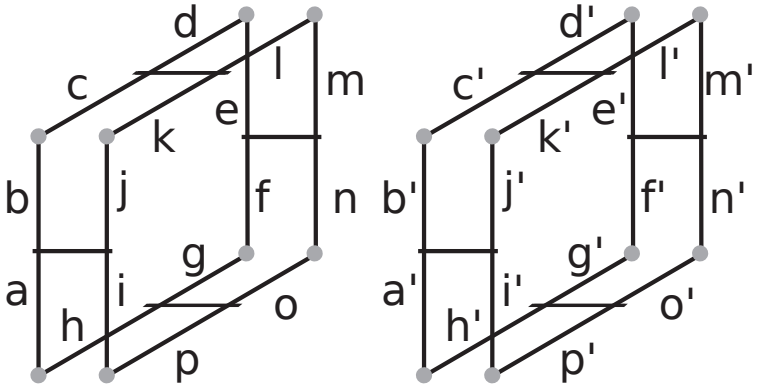
Dans la Fig. 5.6, G et G' sont deux 3G-cartes composées de deux demi-faces reliées par α_3. L'appariement m apparie la demi-face de G composée des brins $\{a, b, c, d, e, f, g, h\}$ à la demi-face de G' composée des brins $\{a', b', c', d', e', f', g', h'\}$. L'appariement du brin j avec le brin j' permet de conserver les coutures $(b, 3, j)$ et $(j, 3, b)$ dans la sous-carte commune. Cependant, la sous-carte extraite n'est pas valide car $\alpha_0 \circ \alpha_3$ n'est pas une involution dans la sous-carte extraite (le brin i n'appartenant pas à la sous-carte extraite). En conséquence, la fonction $Propager$ va tenter de satisfaire cette contrainte en appariant i avec i' ce qui résoud le problème pour $\alpha_0 \circ \alpha_3$. Seulement, l'appariement de i avec i' implique un problème similaire pour l'involution $\alpha_1 \circ \alpha_3$. $Propager$ s'appelle donc récursivement pour apparier les couples de brins requis pour conserver la validité de l'appariement. L'appariement de j avec j' aura finalement engendré l'appariement des deux demi-faces complètes. Maintenant, si au lieu d'apparier les brins j et j', les brins appariés sont j et o', alors la couture en α_3 n'est pas retrouvée. Cela n'implique pas une invalidité de l'appariement car la sous-carte extraite est toujours valide (si l'on considère les sous-cartes induites). La fonction $Propager$ n'apparie donc pas de couples de brins supplémentaires. Notez que la ligne 5 vérifie que $\alpha_i \circ \alpha_j$ soit effectivement une involution dans la sous-carte extraite. Pour cela nous vérifions que l'appariement ainsi construit reste valide en vérifiant la propriété 2 de définition 43.

Après l'appel à la fonction $Propager$, la procédure $ApproxD$ appelle la fonction $estValide^*$ pour tester si l'appariement est valide. En effet, après l'exécution de



b	a	b	c	d
$m(b)$	a'	b'	ϵ	ϵ
$m_1(b)$	a'	b'	c'	ϵ
$m_2(b)$	a'	b'	c'	d'

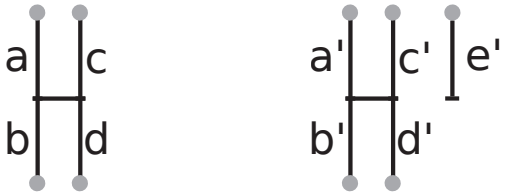
FIGURE 5.5 – *m est un appariement valide, il décrit la sous-carte $G_{\downarrow m}$. L'ajout du couple (c, c') dans m créé m_1 qui est un appariement non valide. En effet, dans $G_{\downarrow m_1}$, $\alpha_0 \circ \alpha_2$ n'est pas une involution. La propagation des contraintes ajoute le couple (d, d') pour obtenir l'appariement valide m_2.*



(a) G (b) G'

b	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
$m(b)$	a'	b'	c'	d'	e'	f'	g'	h'	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ
$m'(b)$	a'	b'	c'	d'	e'	f'	g'	h'	i'	j'	k'	l'	m'	n'	o'	p'

FIGURE 5.6 – *Exemple de deux 3G-cartes G et G' isomorphes composées de deux demi-faces reliées par α_3. m est un appariement entre G et G' qui apparie une demi-face de G avec une demi-face de G'. m' est l'appariement résultant de l'ajout du couple (j, j') à m et d'un appel à $Propager$.*



(a) G (b) G'

b	a	b	c	d
$m(b)$	e'	ϵ	c'	d'

FIGURE 5.7 – *Exemple de deux nG-cartes G et G', et d'un appariement valide m. L'appariement de b avec b' rend m invalide. Il faut apparier a avec a', ce qui n'est pas possible car a est déjà apparié. L'opération d'appariement de b avec b' est donc impossible.*

$Propager$ il existe des cas où l'appariement n'est pas valide. Par exemple dans la Fig. 5.7 l'appariement du brin b avec le brin b' entraine un appariement invalide qui ne peut être corrigé par $Propager$ car il nécessite d'apparier a avec a', alors que a est déjà apparié avec e'. La fonction $estValide^*$ permet de vérifier la validité de l'appariement après la propagation. Pour cela, elle parcourt l'ensemble des brins appariés, et vérifie que pour tout i et j tels que $0 \leq i \leq i+2 \leq j \leq n$, $\alpha_i \circ \alpha_j$ est une involution. Nous avons deux versions de $estValide^*$ pour les deux types de sous-cartes supportés, $estValide^p$ et $estValide^i$. La fonction $estValide^i$ vérifie en plus que les coutures entre deux brins i-cousus appariés sont conservées. Le test pour vérifier qu'un appariement est valide est direct et déduit directement des Définitions 43 et 44.

5.3 Heuristiques pour choisir le prochain couple de brins à apparier

$ApproxD$ construit un appariement en ajoutant successivement des couples de brins choisis dans un ensemble de candidats. Le choix du prochain couple de brins est primordial, car si celui-ci engendre un appariement valide il ne sera jamais remis en cause pour cette exécution de l'algorithme. Le choix des brins est donc dirigé par différentes heuristiques. L'objectif est de construire l'appariement correspondant à une sous-carte commune la plus grande possible. La taille étant dépendante du nombre de brins et du nombre de coutures, ces heuristiques visent à privilégier le choix de couples permettant de retrouver un maximum de coutures, ou encore de privilégier des couples ayant un voisinage similaire, ou un voisinage discriminant.

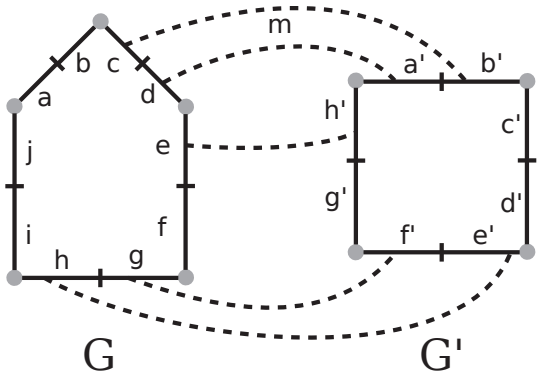
Ces différentes heuristiques pourront ensuite être utilisées conjointement en utilisant un sytème de pondération. Nous présentons tout d'abord une heuristique permettant de retrouver un plus grand nombre de coutures, ensuite deux heuristiques décrivant la structure locale de la nG-carte en utilisant les degrés et co-degrés et enfin une heuristique prenant en compte les étiquettes des nG-cartes.

Heuristique basée sur le nombre de coutures préservées

Pour minimiser la distance décrite par un appariement, l'appariement calculé doit maximiser le nombre de coutures retrouvées. En effet, en retrouvant un maximum de coutures, nous réduirons la probabilité de ne pas pouvoir apparier un couple de brins à cause d'une contrainte d'intégrité. Cela est vrai dans le cas induit, si $\omega_2 = 0$ et d'autant plus vrai si $\omega_2 \neq 0$. L'heuristique de préservation de coutures va orienter l'appariement des couples en ce sens. Elle retourne le nombre de coutures nouvellement préservées pour chaque couple de brins à apparier. Nous ordonnons donc *Cand* pour choisir tout d'abord les couples de brins maximisant la fonction h_{cout} suivante :

$$h_{cout}(b,b',m) = \begin{array}{l} \text{Nombre de coutures de } Extract(G,G',Propager(m,b,b')) \\ - \text{ Nombre de coutures de } Extract(G,G',m) \end{array}$$

Ainsi donné, h_{cout} nécessite le calcul de $Extract(G,G',m)$ à chaque mise à jour de la valeur de l'heuristique. Pour simplifier le calcul effectif, nous le calculons directement à partir de m, b et b' en comptant le nombre de coutures retrouvées sans extraire effectivement la sous-carte commune. On a donc :



brin	m(brin)
c	b'
d	a'
e	h'
g	f'
h	e'

$$Cand = \{a, b, f, i, j\} \times \{c', d', g'\}$$

h_{cout}	a	b	f	i	j
c'	0	2	0	0	0
d'	0	0	0	2	0
g'	0	0	4	0	0

FIGURE 5.8 – *Valeurs de l'heuristique h_{cout} pour tout couple de brins des 2G-cartes G et G' étant donné l'appariement m. Le couple (b, c') permet de retrouver les coutures $(b, 1, c)$ avec $(b', 1, c')$ et leurs symétriques. Le couple (i, d') permet de retrouver les coutures $(i, 1, h)$ avec $(d', 1, e')$ et leurs symétriques. Le couple (f, g') permet de retrouver les 4 coutures $(f, 1, g), (g, 1, f), (f, 0, e), (e, 0, f)$ avec les coutures $(f', 1, g'), (g', 1, f'), (h', 0, g'), (g', 0, h')$.*

$$h_{cout}(b, b', m) = \sum_{0 \leq i \leq n} (\delta_{cout}(b, b', m, i)) \tag{5.1}$$

où $(b, b') \in Cand, m$ est l'appariement en cours de construction et

$$\begin{cases} \delta_{cout}(b, b', m, i) = 2 & \text{si b et b' sont i-cousus à 2 brins qui sont déjà apparié entre eux,} \\ & \text{i.e. } \alpha_i(b) \neq b, \alpha'_i(b') \neq b', m(\alpha_i(b)) \neq \epsilon \text{ et } m(\alpha_i(b)) = \alpha'_i(b')); \\ \delta_{cout}(b, b', m, i) = 0 & \text{sinon .} \end{cases}$$

Le nombre de coutures retrouvées est toujours un nombre pair, car si nous retrouvons la couture (b, i, b') alors nous retrouvons aussi la couture (b', i, b).

La Fig. 5.8 illustre l'heuristique $h_{cout}(b, b', m)$. Les brins de la 2G-carte G sont appariés avec les brins de la 2G-carte G' par l'appariement m. L'heuristique nous indique, pour chaque couple, le nombre de coutures retrouvées. Pour maximiser le nombre de coutures retrouvées, il est plus intéressant de commencer par apparier des couples de brins qui retrouvent le plus de coutures, i.e., f et g' sur cet exemple.

La valeur de l'heuristique pour chaque couple de brins est amenée à être modifiée après chaque appariement. La méthode de mise-à-jour naïve parcourt à chaque fois tous les couples de *Cand* pour recalculer la valeur de l'heuristique. Au premier couple de brins, nous allons calculer $|B| \cdot |B'|$ valeurs, puis de nouveau toutes les valeurs des brins ne contenant pas les brins appariés, soit $(|B| - 1) \cdot (|B'| - 1)$ après le premier appariement et ainsi de suite. Dans le cas où tous les brins sont appariés, nous aurons une complexité de l'ordre de $min(|B|, |B'|) \cdot |B| \cdot |B'|$.

Nous avons amélioré cette complexité en proposant une structure utilisant conjointement un tas binaire et une forme matricielle. Nous stockons l'ensemble des valeurs de h_{cout} associés aux couples de *Cand* dans une matrice de $|B| \times |B'|$ cases. Cette matrice nous permet de ne pas parcourir l'ensemble *Cand* en entier pour mettre à jour les

valeurs. En effet, dans le cas de l'appariement du couple (b, b'), l'ensemble des cases à invalider sont sur la ligne correspondant au brin b et la colonne correspondant au brin b'. L'ensemble des valeurs des heuristiques à modifier sont sur les cases de lignes $\alpha_i(b)$ et de colonnes $\alpha_i(b')$. Ainsi, pour l'appariement d'un couple de brins, nous parcourons $|B| + |B'|$ cases à invalider correspondant aux brins appariés et $n + 1$ cases sont mises à jour.

Pour choisir le meilleur couple dans l'ensemble *Cand*, la méthode naïve consiste encore une fois à parcourir entièrement la matrice. La structure matricielle proposée pour améliorer la complexité de la mise à jour ne permet pas d'extraire le meilleur élément rapidement. Nous utilisons donc conjointement un tas binaire permettant d'extraire en temps logarithmique la case ayant la valeur de l'heuristique la plus élevée. Ainsi, la complexité associée à la recherche de la meilleure valeur dans la matrice est de $log(|B| \cdot |B'|)$. Seulement pour maintenir le tas binaire, nous devons ajouter le coût de mise à jour du tas à la complexité de mise à jour de nos cases matricielles. La complexité de mise à jour d'un élément dans un tas binaire étant dans le pire des cas de $log(|B| \cdot |B'|)$, la complexité de la phase de mise à jour après appariement devient donc $(|B| + |B'| + n) \cdot log(|B| \cdot |B'|)$. La complexité effective est bien évidemment moindre, car nous retirons de notre tas chacun des couples ne pouvant plus être choisi (ayant au moins 1 brin apparié), ainsi la complexité associée à la mise à jour diminue au fur et à mesure de l'appariement des brins.

Degrés et Co-degrés

L'heuristique h_{cout} permet d'étendre un appariement existant en appariant de nouveaux brins tout en commençant par apparier les brins permettant de retrouver le plus de coutures. Elle ne donne par contre pas d'indication concernant le premier couple de brins à apparier qui pourtant est primordial. Nous proposons ici une seconde heuristique permettant de choisir d'apparier des brins ayant un voisinage similaire.

Nous savons que dans une nG-carte tout brin est incident à $n + 1$ cellules, nous allons utiliser l'information découlant de ces cellules pour décrire le voisinage du brin. Pour cela nous utilisons les degrés et les co-degrés des i-cellules, donnant respectivement le nombre de $(i-1)$-cellules et de $(i+1)$-cellules distinctes incidentes à la i-cellule concernée. Par exemple, le co-degré d'une face nous indique le nombre d'arêtes incidentes à la face. Nous pouvons ainsi espérer que si nous apparions deux brins appartenant à deux faces ayant le même co-degré, nous puissions apparier tous les brins de cette face. Un raisonnement similaire peut être tenu avec toutes les cellules. Une première heuristique nommée h_{deg} est donc basée sur les degrés et co-degrés des cellules des deux brins.

$$h_{deg}(b, b') = \sum_{0 \leq i \leq n} (\delta_{deg}(b, b', i) + \delta_{codeg}(b, b', i)) \tag{5.2}$$

Où
- $\delta_{deg}(b, b', i) = 1$ si le degré de la i-cellule de b est égal au degré de la i-cellule de b' ;
- $\delta_{deg}(b, b', i) = 0$ sinon ;
- $\delta_{codeg}(b, b', i) = 1$ si le co-degré de la i-cellule de b est égal au co-degré de la i-cellule de b' ;
- $\delta_{codeg}(b, b', i) = 0$ sinon ;

Une seconde heuristique nommée h^+_{deg} intègre en plus l'information de la cellule adjacente, si elle existe, pour une description plus précise. Pour cela nous regardons si

le brin b est i-cousu, si tel est le cas alors nous comparons les degrés et co-degrés de la i-cellule du brin $\alpha_i(b)$ avec la i-cellule du brin i-cousu (s'il existe) à b'.

$$h_{deg+}(b,b') = h_{deg}(b,b') + \sum_{0\leq i\leq n} (\delta^+_{deg}(b,b',i) + \delta^+_{codeg}(b,b',i)) \tag{5.3}$$

Où
- $\delta^+_{deg}(b,b',i) = 1$ si b et b' sont i-cousus et que le degré de la i-cellule de $\alpha_i(b)$ est égal au degré de la i-cellule de $\alpha'_i(b')$;
- $\delta^+_{deg}(b,b',i) = 0$ sinon ;
- $\delta^+_{codeg}(b,b',i) = 1$ si b et b' sont i-cousus et que le co-degré de la i-cellule de $\alpha_i(b)$ est égal au co-degré de la i-cellule de $\alpha'_i(b')$;
- $\delta^+_{codeg}(b,b',i) = 0$ sinon .

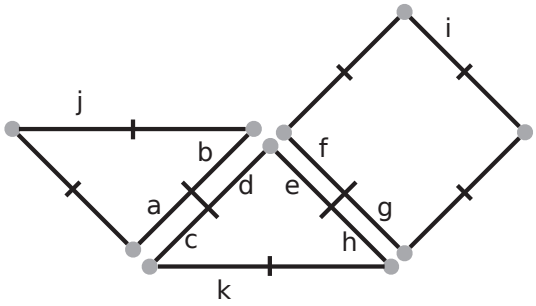
$h_{deg}(b,b')$ est bornée entre 0 et $2n+2$, et donne le nombre de degrés et co-degrés identiques des cellules incidentes aux deux brins comparés. $h_{deg+}(b,b')$ est bornée entre 0 et $4n+4$, et donne le nombre de degrés et co-degrés similaires des cellules incidentes et adjacentes aux deux brins comparés.

La Fig. 5.9 illustre l'heuristique basée sur les degrés et co-degrés des cellules incidentes et adjacentes aux brins. Si nous étudions l'heuristique h_{deg} sur cet exemple, nous notons que 13 couples de brins du sous-ensemble de brins considéré obtiennent la valeur maximale de 4, ce qui signifie que nous ne savons pas différencier les meilleurs d'entre eux. Il est alors possible d'améliorer cette heuristique en tenant compte en plus des degrés et co-degré des cellules des brins, des degrés et co-degrés des cellules adjacentes à la cellule du brin avec l'heuristique h^+_{deg}. Le couple (d,a') est le seul couple pour le sous-ensemble de brins considéré qui obtient le score maximal de 8. Si nous étudions ce couple, les deux brins font partie de triangles, qui ont pour face adjacente des triangles, leurs sommets ainsi que les sommets adjacents sont incidents à 3 arêtes, leurs arêtes et arêtes adjacentes sont incidentes à 2 faces. Cette heuristique va donc donner un point de départ de l'algorithme de construction d'un appariement qui va nous permettre par propagation d'apparier au minimum les deux triangles et leurs triangles adjacents. Considérons maintenant les brins, i, k, j et c'. Ces brins n'ont pas de face adjacente et sont donc pénalisés dans le calcul de l'heuristique car ils ne pourront jamais retrouver le co-degré de la 2-cellule adjacente. Cependant, le choix de ces couples de brins ne garantit pas d'apparier plus que leur face, contre deux faces complètes pour le précédent choix.

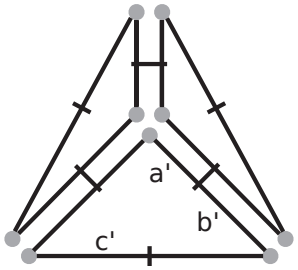
Considérons maintenant une 2G-carte représentant un plan téssélé, uniquement composé de triangles. La principale difficulté du problème de comparaison de cette 2G-carte avec une autre 2G-carte est que l'heuristique h_{deg+} sera moins pertinente.

Cette difficulté est illustrée dans la Fig. 5.10 où tous les brins sont incidents à des 2-cellules triangulaires. Le co-degré de chaque face est donc de 3, le co-degré des faces adjacentes, s'il y en a, est aussi de 3. Nous avons donc une 2G-carte pour laquelle les co-degrés n'apportent pas d'information. Il est possible de générer des cas avec encore plus de similarités, un exemple extrême étant la grille régulière torique où tous les degrés et tous les co-degrés sont identiques.

Dans un maillage triangulaire beaucoup de couples candidats auront la valeur heuristique maximale. Nous proposons donc d'affiner cette heuristique en commençant par les brins ayant les degrés et co-degrés les moins présents dans la nG-carte, l'objectif étant de commencer par les points les plus particuliers, et donc éventuellement les plus intéressants.



(a) G



(b) G'

brin	a	b	c	d	e	f	g	h	i	j	k	a'	b'	c'
degree(0-cellule($brin$))	3	4	3	4	4	4	3	3	2	2	3	3	3	3
degree(1-cellule($brin$))	2	2	2	2	2	2	2	2	1	1	1	2	2	1
co-degree(1-cellule($brin$))	2	2	2	2	2	2	2	2	2	2	2	2	2	2
co-degree(2-cellule($brin$))	3	3	3	3	3	4	4	3	4	3	3	3	3	3
degree(0-cellule($\alpha_0(brin)$))	4	3	4	3	3	3	4	4	2	3	3	3	3	3
degree(1-cellule($\alpha_1(brin)$))	1	1	1	2	2	1	1	1	1	1	2	2	1	2
co-degree(1-cellule($\alpha_1(brin)$))	2	2	2	2	2	2	2	2	2	2	2	2	2	2
co-degree(2-cellule($\alpha_2(brin)$))	3	3	3	3	4	3	3	4				3	3	

(c) Tableaux des degrés et co-degrés

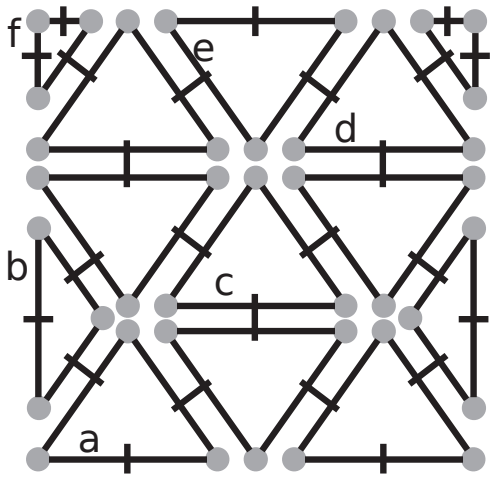
	a	b	c	d	e	f	g	h	i	j	k
a'	4	3	4	3	3	2	3	4	1	2	3
b'	4	3	4	3	3	2	3	4	1	2	3
c'	3	2	3	2	2	1	2	3	2	3	4

(d) Tableau heuristique h_{deg}

	a	b	c	d	e	f	g	h	i	j	k
a'	6	6	6	7	6	6	5	5	2	3	6
b'	7	7	7	6	5	6	6	6	3	4	5
c'	4	4	4	5	5	3	3	4	3	4	7

(e) Tableau heuristique h_{deg+}

FIGURE 5.9 – *G et G' sont deux 2G-cartes. Le tableau (c) donne les valeurs de degrés et co-degrés nécessaires au calcul de l'heuristique. Le tableau (d) donne pour chaque couple de brins (b, b'), les valeurs de $h_{deg}(b, b')$ sans tenir compte des cellules adjacentes et le tableau (e) donne les valeurs de $h^+_{deg}(b, b')$ en tenant compte des cellules adjacentes.*



(a) Carte d'un Plan tessélé

$brin$	a	b	c	d	e	f
degree(0-cellule($brin$))	3	5	5	6	4	2
degree(1-cellule($brin$))	1	1	2	2	2	1
co-degree(1-cellule($brin$))	2	2	2	2	2	2
co-degree(2-cellule($brin$))	3	3	3	3	3	3
degree(0-cellule($\alpha_0(brin)$))	4	3	5	5	6	5
degree(1-cellule($\alpha_1(brin)$))	2	2	2	2	1	1
co-degree(1-cellule($\alpha_1(brin)$))	2	2	2	2	2	2
co-degree(2-cellule($\alpha_2(brin)$))			3	3	3	

(b) Tableaux des degrés et co-degrés

FIGURE 5.10 – *(a) Une 2G-carte représentant un plan tessélé. Le tableau (b) donne les valeurs de degrés, et co-degrés nécessaires au calcul de h_{deg}. Dans le cas de maillages régulier les degrés et co-degrés sont moins discriminants.*

Nous caractérisons chaque brin par l'histogramme de ses degrés et co-degrés.

$$tuple(b) = (degree(b,0), \ldots, degree(b,n-1), codegree(b,1), , \ldots, codegree(b,n)) \quad (5.4)$$

$$\text{Où} \left| \begin{array}{l} degree(b,i) = degree(i - cellule(b)) \\ codegree(b,i) = codegree(i - cellule(b)) \end{array} \right.$$

Nous étendons ce principe en intégrant les degrés et co-degrés des cellules voisines :

$$\begin{aligned} tuple^{+}(b) =& (degree(b,0), \ldots, degree(b,n-1), codegree(b,1), , \ldots, codegree(b,n), \\ & degree^{+}(b,0), \ldots, degree^{+}(b,n-1), codegree^{+}(b,1), , \ldots, codegree^{+}(b,n)) \end{aligned} \quad (5.5)$$

$$\text{Où} \left| \begin{array}{ll} degree^{+}(b,i) = degree(i - cellule(\alpha_i(b))) & \text{si } \alpha_i(b) \neq b; \\ degree^{+}(b,i) = 0 & \text{sinon ;} \\ codegree^{+}(b,i) = codegree(i - cellule(\alpha_i(b))) & \text{si } \alpha_i(b) \neq b; \\ codegree^{+}(b,i) = 0 & \text{sinon .} \end{array} \right.$$

Par exemple dans la Fig. 5.10, nous aurons $tuple(c) = (5,2,2,3)$, et $tuple(a) = (3,1,2,3)$. Si nous prenons en compte les cellules adjacentes, nous utiliserons $tuple^{+}(c) = (5,2,2,3,5,2,2,3)$ et $tuple^{+}(a) = (3,1,2,3,4,2,2,0)$. Ce tuple décrit l'ensemble des degrés et co-degrés de toutes les cellules du brin concerné. C'est donc une description intéressante de son voisinage. L'objectif est donc de trouver les brins ayant des voisinages particuliers. Nous choisissons des couples de brins ayant le même tuple, en privilégiant ceux qui sont les plus rares. En effet, dans un maillage triangulaire, on ne peut apparier que des brins appartenant à des triangles. Si par contre il existe un quadrilatère parmi les triangles dans les deux 2G-cartes, il peut être un point de départ plus intéressant.

Nous définissons la fréquence du tuple associé à un brin b.

$$\phi(b) = |\{b' \in B | tuple(b) = tuple(b')\}| \quad (5.6)$$

Nous en déduisons la fréquence maximum d'un tuple dans l'ensemble des tuples de la nG-carte.

$$\phi_{max}(G) = \max_{b \in B}(\phi(b)) \quad (5.7)$$

Enfin nous définissons la valeur de l'heuristique comme la fréquence minimum des tuples des deux brins s'ils sont identiques.

$$\begin{aligned} h_{freq}(b,b') =& min(\{\phi_{max}(G), \phi_{max}(G')\}) - min(\{\phi(b), \phi(b')\}) \text{ si } tuple(b) = tuple(b'); \\ h_{freq}(b,b') =& 0 \text{ sinon.} \end{aligned} \quad (5.8)$$

Si la nG-carte possède beaucoup de faces similaires, alors la fréquence du tuple ne sera pas pertinente car trop de tuples auront la même valeur. Nous pourrons alors utiliser $tuple^{+}(b)$. Au contraire si la nG-carte est très diversifiée (en termes de degrés et co-degrés de cellules), alors l'utilisation de $tuple^{+}(b)$ ne permettra pas de retrouver deux brins ayant le même tuple. Le choix de l'utilisation de $tuple(b)$ ou de $tuple^{+}(b)$ dépend donc des nG-cartes comparées, et donc des objets modélisés par ces nG-cartes.

Similarité des étiquettes

Nous avons vu que la distance d'édition peut être étendue pour intégrer de l'information supplémentaire en fonction des objets modélisés en utilisant des étiquettes. Il est alors possible d'intégrer dans l'heuristique du choix du prochain couple de brins l'information portée par les étiquettes.

Si G et G' sont deux nG-cartes étiquetées, alors l'heuristique associée à l'appariement de deux brins b et b' est $h_{etiq}(b, b')$.

$$h_{etiq}(b, b') = cost(e(b), e(b')) \tag{5.9}$$

Où $cost(l_1, l_2)$ est le coût de substitution de l'étiquette l_1 par l'étiquette l_2. Notez que pour cette heuristique, le choix est d'autant meilleur que sa valeur est petite.

Cette heuristique va permettre d'orienter la construction de l'appariement en commençant par mettre en relation des brins ayant des étiquettes proches.

Composition des Heuristiques

Toutes ces heuristiques visent à choisir le meilleur couple de brins. L'heuristique h_{cout} guide le choix d'un nouveau couple en fonction du nombre de coutures retrouvées. En cas d'égalité (par exemple lors du choix du premier couple), nous utilisons une combinaison des heuristiques $h_{deg}, h^{+}_{deg}, h_{freq}$ et h_{etiq} pour départager les ex-aequos. Pour cela nous proposons un ensemble de poids $\theta_1, \theta_2, \theta_3$ et θ_4 pour pondérer l'importance de chacune des heuristiques.

$$h_{choix}(b, b') = \theta_1.h_{deg}(b, b') + \theta_2.h_{deg^+}(b, b') + \theta_3.h_{freq}(b, b') + \frac{\theta_4}{h_{etiq}(b, b')} \tag{5.10}$$

5.4 Évaluation expérimentale de ApproxD

Dans cette section nous souhaitons valider notre algorithme et montrer que les approximations de la distance calculées sont proches de la distance exacte. Nous allons aussi étudier l'impact des différentes heuristiques proposées. Pour cela nous proposons un ensemble de quatre jeux de données $Taille_x, Noyau_x, Triangles$ et $Bornes$.

Jeux d'essais considérés

$Taille_x$ avec $x \in [500; 20000]$: Pour ce jeu d'essai nous avons généré des couples de 3G-cartes aléatoirement de tailles différentes. Pour cela nous avons ajouté successivement des brins à une 3G-carte vide. L'ajout d'un brin aléatoirement est réalisé en sélectionnant aléatoirement une dimension i puis un brin i-libre. Nous ajoutons un nouveau brin en le i-cousant au brin sélectionné. Nous avons vu précédemment dans ce chapitre que l'ajout d'un brin unique dans une nG-carte peut entraîner des problèmes d'intégrité. Ainsi, pour chaque ajout de brins nous ajoutons d'autres brins et coutures permettant de résoudre les contraintes violées. Nous avons généré ici 159 paires de 3G-cartes ayant une taille variant de 250 brins à 20000 avec un pas de 125 brins.

$Noyaux_x$avec $x \in \{1000, 2500, 4000\}$: Pour ce jeu, nous avons généré des couples de 3G-cartes ayant une partie commune connue ce qui nous permet d'une part de

contrôler la similarité des 3G-cartes comparées, et d'autre part de connaître une borne minimale sur la taille de la plus grande sous-carte commune. La partie commune entre deux 3G-cartes est appelée *noyau*. La création de ces 3G-cartes débute par la création du noyau en ajoutant successivement à une 3G-carte vide des brins et en les cousant aléatoirement avec le même procédé que pour le jeu $Taille_x$. Une fois que le noyau a la taille désirée, nous le dupliquons et nous traitons chaque copie séparément en ajoutant des brins aléatoirement jusqu'à obtenir des 3G-cartes de la taille désirée. Les 3G-cartes générées dans ce jeu de données ont 5000 brins, et un noyau de 1000, 2500 et 4000 brins ce qui correspond à une proportion de noyau de 20%, 50% et 80%. Pour chaque taille de noyau nous avons généré 50 paires de cartes.

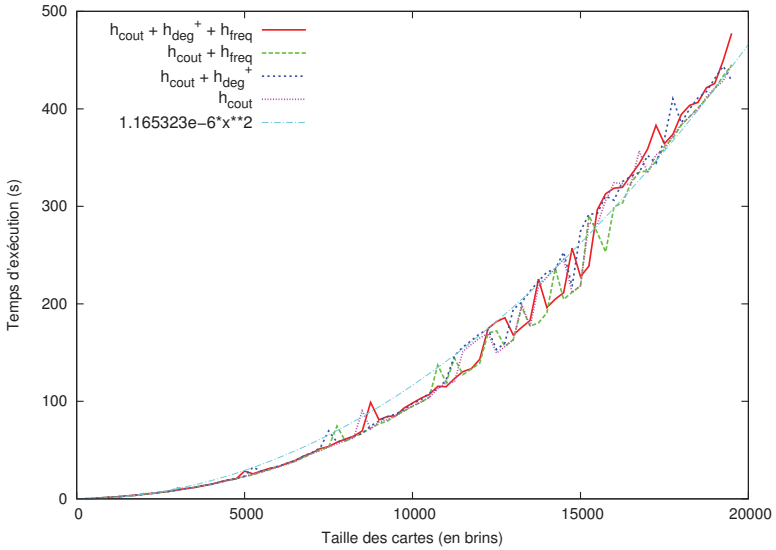
$Triangles$: Dans ce jeu de données, nous avons cherché à maximiser la ressemblance entre les 2G-cartes comparées. Nous avons proposé dans la section précédente une heuristique basée sur les degrés et co-degrés des i-cellules des nG-cartes moins discriminante pour des nG-cartes similaires. Nous avons donc généré 50 paires de 2G-cartes de 5000 brins composées uniquement de triangles. Nous avons utilisé un procédé de génération proche des jeux précédents. Seulement, au lieu d'ajouter successivement des brins, nous ajoutons des ensembles de 6 brins cousus pour former un triangle. Nous 2-cousons ensuite aléatoirement une arête de ce triangle à une arête 2-libre du maillage. Pour maximiser la similarité et ainsi augmenter la difficulté, les 2G-cartes comparées ont un noyau commun de 95%.

$Bornes$: Pour ce jeu nous souhaitons pouvoir comparer l'approximation de la distance calculée par notre algorithme pour un ensemble de 2G-cartes pour lesquelles nous connaissons la valeur de la distance $d^p_{1,1}$. Pour connaître la valeur de la distance nous procédons par construction en dégradant avec les opérations del et $unsew$ des 2G-cartes pour générer des sous-cartes partielles. L'ensemble des opérations utilisées pour modifier les 2G-cartes nous donne un chemin d'édition pour passer de la 2G-carte originale à la sous-carte partielle. En conséquence il est possible de calculer la valeur exacte de la distance $d^p_{1,1}$. Nous comparons ainsi les distances approximées $d^i_{1,1}$ et $d^p_{1,1}$ avec la distance exacte $d^p_{1,1}$. Nous notons la 2G-carte initiale G_0 et nous générons un ensemble de 14 sous-cartes partielles G_i en supprimant aléatoirement $i \times 5\%$ des brins ou coutures de G_0. Nous avons considéré un ensemble de 60 cartes, composées de 15 cartes construites à partir de 4 2G-cartes initiales modélisant des maillages de la base [AIM]. Les 2G-cartes originales ont en moyenne 4808 brins et 14031 coutures.

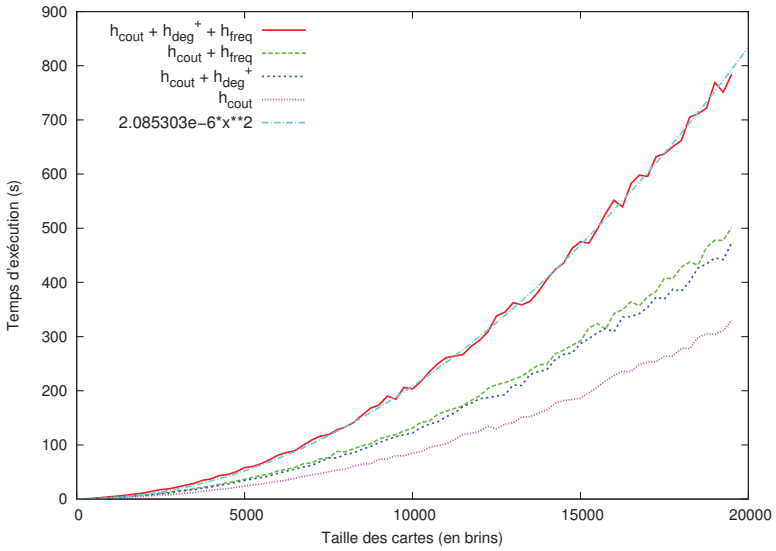
Évaluation expérimentale des temps d'exécution sur $Taille_x$

L'objectif de cette évaluation est d'étudier le passage à l'échelle de notre algorithme en faisant varier la taille de différentes nG-cartes comparées. Nous séparons le temps d'initialisation de la structure du temps d'exécution étant donné que pour une seule initialisation nous pouvons réaliser autant d'éxécutions de l'algorithme que nous le souhaitons. En effet, chacune de ces exécutions peut choisir un couple de brins différent lors du choix du premier couple à apparier, de sorte qu'elles construiront des appariements différents. Nous pouvons alors ne garder que l'appariement induisant la meilleur approximation.

Dans la Fig. 5.11 nous remarquons que le temps d'initialisation dépend bien du nombre de brins des 3G-cartes. C'est normal, car nous devons pour chaque couple de



(a) Temps d'exécution de ApproxD



(b) Temps d'exécution de l'initialisation

FIGURE 5.11 – *Étude du temps d'exécution de $ApproxD$ en fonction de la taille des 3G-cartes comparées. Le temps d'exécution d'un appel à $ApproxD$ suit une fonction quadratique quand le nombre de brins des 3G-cartes comparées augmente. Cependant, le temps d'exécution d'un glouton ne varie pas en fonction de l'heuristique utilisée. Dans le second graphique nous étudions le temps d'initialisation de l'algorithme avant exécution. Nous notons qu'il augmente aussi quadratiquement en fonction du nombre de brins des 3G-cartes comparées et aussi en fonction des heuristiques utilisées.*

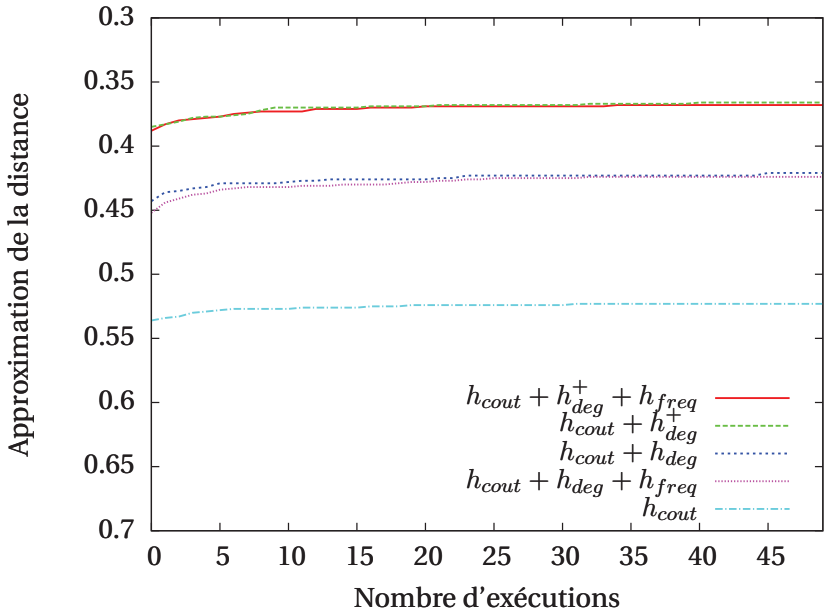


FIGURE 5.12 – *Évaluation des heuristiques pour approximer la distance $d^{i,1,1}_{mcs}$ sur le jeu de données $Noyau_{1000}$, soit un noyau de 20%. Pour chaque nombre d'exécution $x \in [1;50]$, les courbes donnent la plus petite approximation calculée sur les x exécutions (en moyenne pour les 50 couples de 3G-cartes du jeu de données.)*

brins calculer la valeur des heuristiques utilisées et l'insérer dans le tas binaire. Notez que lorsque la taille des 3G-cartes augmente, nous augmentons simultanément les deux 3G-cartes comparées. En utilisant la méthode des moindre carrés, nous notons qu'il est possible d'associer des fonctions quadratiques à chacune de nos courbes, ce qui valide bien le fait que l'algorithme est de complexité quadratique en pratique.

Évaluation expérimentale des heuristiques sur $Noyaux_x$ et $Triangles$

Pour valider nos heuristiques et l'algorithme, nous avons généré des 2G-cartes et 3G-cartes avec une partie commune (sous-carte). Cela nous permet d'étudier le comportement des heuristiques sur des paires de 2G-cartes plus ou moins similaires.

Dans les figures 5.12,5.13 et 5.14, nous calculons une approximation supérieure de $d^{i,1,1}_{mcs}$ et dans les figures 5.15,5.16 et 5.17, nous calculons une approximation supérieure de $d^{p,1,1}_{mcs}$ sur le même jeu de données. Les différentes heuristiques comparées sont h_{cout} qui permet de retrouver les coutures, h_{deg} qui décrit la répartition des degrés et co-degrés des cellules des brins comparés et enfin h^+_{deg} qui prend en compte les degrés et co-degrés des cellules adjacentes. Pour chaque heuristique nous exécutons 50 fois l'algorithme $ApproxD$ en choisissant un couple de départ différent pour chaque exécution. Les 50 couples de brins de départ choisis sont les 50 couples ayant la meilleure valeur pour l'heuristique.

La première observation que nous pouvons faire est que quel que soit le type de sous-carte considéré, et quelle que soit la taille du noyau lors de la génération, l'heuristique h^+_{deg} permet de trouver une meilleure approximation de notre distance que les heuristique h_{deg} et h_{cout} seules. La même remarque peut être faite entre l'heuristique h_{deg} et l'heuristique h_{cout} seules. En effet, l'utilisation de h_{cout} uniquement ne permet pas d'obtenir des approximations acceptables. Cela confirme le fait que le choix du pre-

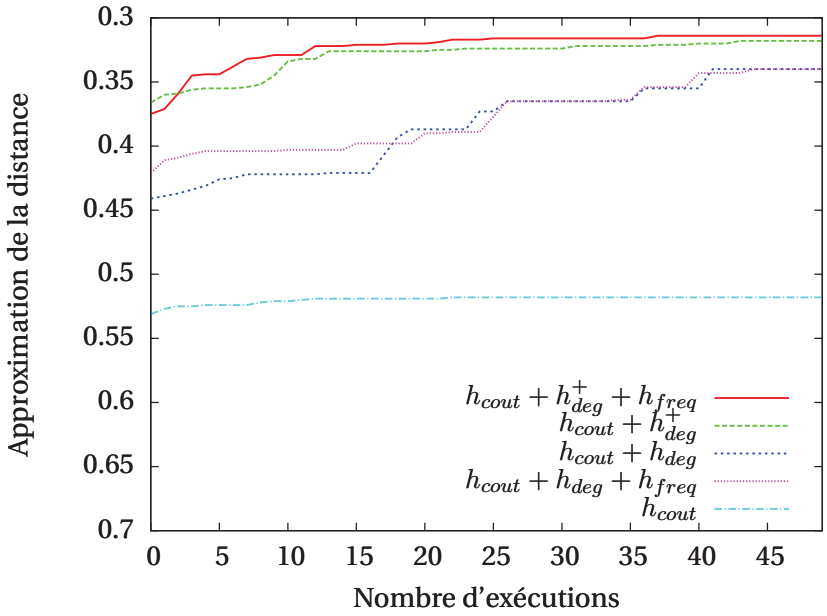


FIGURE 5.13 – *Évaluation des heuristiques pour approximer la distance $d^{i,1,1}_{mcs}$ sur le jeu de données $Noyau_{2500}$, soit un noyau de 50%. Pour chaque nombre d'exécutions $x \in [1;50]$, les courbes donnent la plus petite approximation calculée sur les x exécutions (en moyenne pour les 50 couples de 3G-cartes du jeu de données.)*

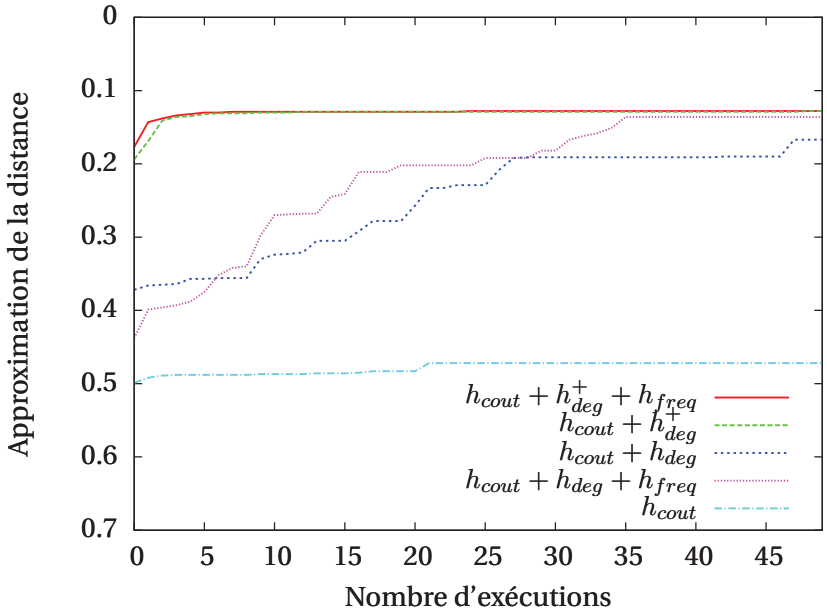


FIGURE 5.14 – *Évaluation des heuristiques pour approximer la distance $d^{i,1,1}_{mcs}$ sur le jeu de données $Noyau_{4000}$, soit un noyau de 80%. Pour chaque nombre d'exécutions $x \in [1;50]$, les courbes donnent la plus petite approximation calculée sur les x exécutions (en moyenne pour les 50 couples de 3G-cartes du jeu de données.)*

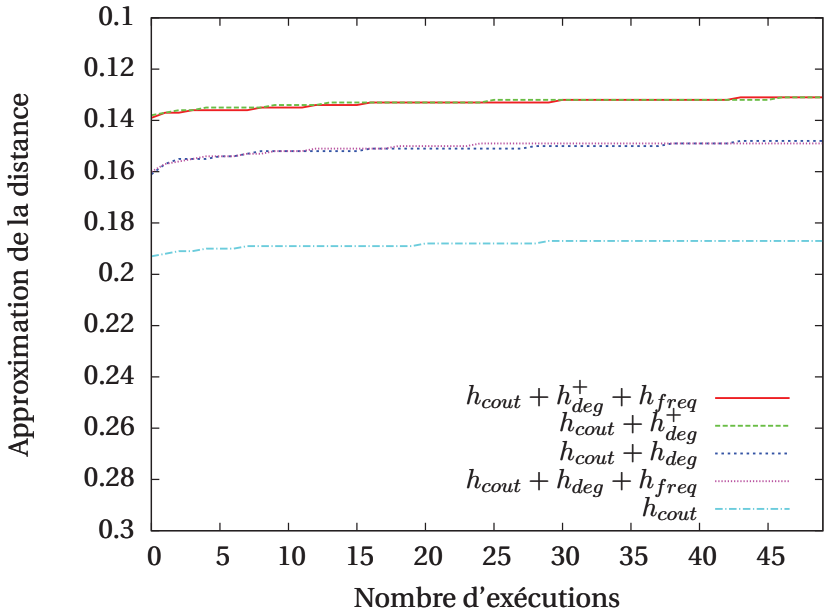


FIGURE 5.15 – *Évaluation des heuristiques pour approximer la distance $d^{p,1,1}_{mcs}$ sur le jeu de données $Noyau_{1000}$, soit un noyau de 20%. Pour chaque nombre d'exécutions $x \in [1;50]$, les courbes donnent la plus petite approximation calculée sur les x exécutions (en moyenne pour les 50 couples de 3G-cartes du jeu de données.)*

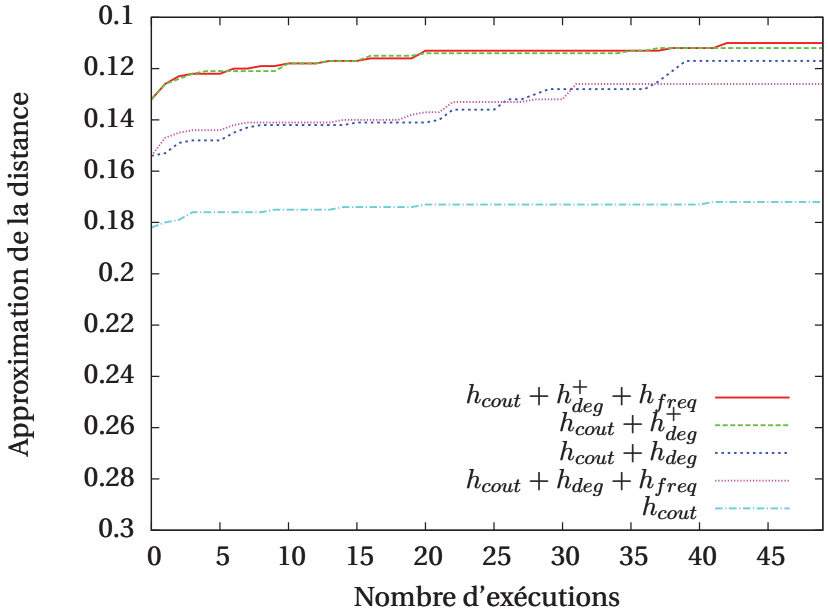


FIGURE 5.16 – *Évaluation des heuristiques pour approximer la distance $d^{p,1,1}_{mcs}$ sur le jeu de données $Noyau_{2500}$, soit un noyau de 50%. Pour chaque nombre d'exécutions $x \in [1;50]$, les courbes donnent la plus petite approximation calculée sur les x exécutions (en moyenne pour les 50 couples de 3G-cartes du jeu de données.)*

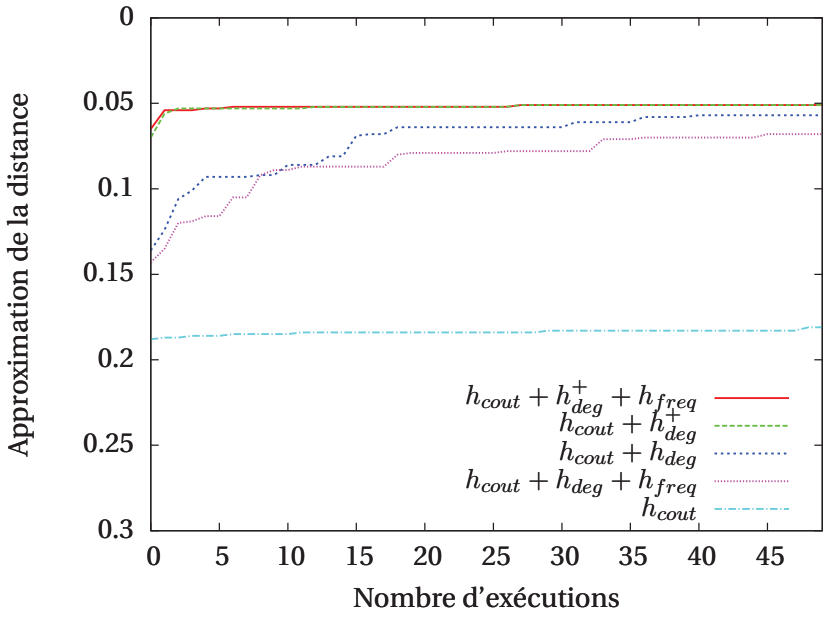
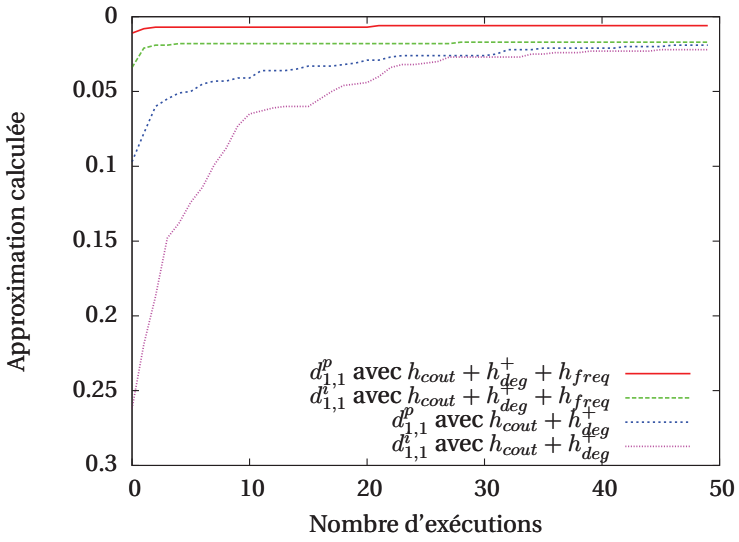


FIGURE 5.17 – *Évaluation des heuristiques pour approximer la distance $d^{p,1,1}_{mcs}$ sur le jeu de données $Noyau_{4000}$, soit un noyau de 80%. Pour chaque nombre d'exécutions $x \in [1;50]$, les courbes donnent la plus petite approximation calculée sur les x exécutions (en moyenne pour les 50 couples de 3G-cartes du jeu de données.)*

mier couple de brins est primordial. Notez que le processus de génération aléatoire produit naturellement des nG-cartes possédant des degrés et co-degrés très variés, ce qui rend les heuristiques h_{deg} et h^+_{deg} très discriminantes. Cette diversité est justement ce qui fait que l'heuristique h_{freq} n'apporte pas de gains significatifs car elle est destinée à améliorer les cas où les degrés et co-degrés sont très similaires. Il est aussi intéressant de regarder l'ordre de grandeur des distances en fonction du noyau et du type de sous-carte considéré. Si nous regardons les distances calculées avec la sous-carte induite, nous observons qu'elles sont égales à 0.37 pour un noyau de 20% et à 0.11 pour un noyau de 80%. Notez que nous recherchons une plus grande sous-carte commune non connexe ce qui signifie que l'algorithme construit dans le meilleur des cas un appariement qui apparie la sous-carte commune du noyau et aussi de nombreux sous-ensembles de brins parmi les brins ajoutés aléatoirement. Pour la sous-carte partielle, les distances calculées sont égales à 0.13 pour le noyau de 20% et à 0.05 pour un noyau de 80%. La différence avec la sous-carte induite est inhérente au fait que la sous-carte partielle permet de retrouver des brins même si leurs coutures ne sont pas retrouvées.

L'heuristique h_{freq} étant destinée à choisir un couple de brins discriminant quand les degrés et co-degrés sont peu variés, nous avons créé le jeu de données $Triangles$ pour lequel nous avons volontairement introduit une très forte similarité. La Fig. 5.18 illustre les résultats. Nous notons que l'heuristique permet d'obtenir de meilleurs résultats plus rapidement indépendamment du type de sous-carte considéré. Elle est donc utile si nous souhaitons comparer des nG-cartes ayant des i-cellules de degrés peu variés.



(a) Sous-carte commune de 95%

FIGURE 5.18 – *Comparaison de l'approximation de la distance calculée en fonction du nombre d'exécutions de l'algorithme. Pour chaque nombre d'exécution $x \in [1;50]$, les courbes donnent la plus petite approximation calculée sur les x exécution (en moyenne pour les 50 couples de 2G-cartes du jeu de données. Notez que dans ce cas, l'heuristique basée sur la fréquence apporte un gain significatif, que la sous-carte considérée soit induite ou partielle.*

Évaluation expérimentale de la qualité de l'approximation sur $Bornes$

Dans cette expérimentation nous souhaitons mettre en évidence la qualité de la distance approximée sur le jeu de données $Bornes$ pour lequel nous connaissons la distance exacte $d^{p,1,1}_{mcs}$. Nous utilisons ici les heuristiques h_{cout}, h^{+}_{deg} et h_{freq}. La Fig. 5.19 montre que le résultat de l'approximation de nos distances est plutôt proche de la distance exacte. Elle montre aussi que l'approximation $d^{p,1,1}_{mcs}$ est plus proche de sa valeur exacte que l'approximation de $d^{i,1,1}_{mcs}$. C'est la conséquence directe de l'utilisation de l'opération $unsew$ dans la modification de la carte originale. La 2G-carte ainsi modifiée est une sous-carte partielle mais n'est pas une sous-carte induite car les involutions entre les brins de la sous-carte ne sont alors plus conservées. Il existe donc des brins appartenant à la sous-carte partielle ne pouvant pas être conservés dans la sous-carte induite car leurs coutures ne sont pas préservées.

5.5 Discussion

Nous avons présenté un algorithme permettant de calculer une approximation supérieure des distances définies dans le Chap. 4. Nous calculons ces distances en calculant un appariement de brins représentant une sous-carte commune. Le calcul de cet appariement est orienté par des heuristiques permettant de prendre en compte la structure des cellules à proximité des brins et aussi l'information que nous pouvons ajouter aux brins au moyen d'étiquettes. La distance calculée est donc dépendante des paramètres utilisés pour les heuristiques, et de la structure des nG-cartes que nous souhaitons comparer. Nous avons évalué l'impact de ces heuristiques sur le calcul des distances $d^{*,1,1}_{mcs}$. Ces heuristiques sont une base permettant de comparer n'importe quel type de nG-carte indépendamment de l'objet modélisé. Il serait sans doute plus effi-

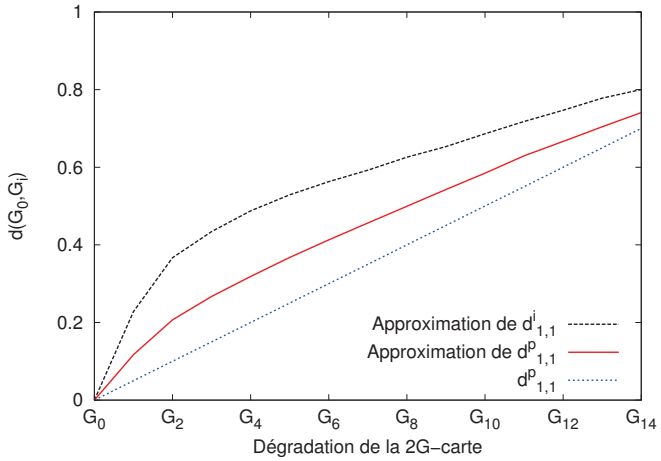


FIGURE 5.19 – *Comparaison de $d^p_{1,1}$ avec les meilleurs approximations de $d^p_{1,1}$ et $d^i_{1,1}$ calculées par 10 exécutions de l'algorithme $ApproxD$ avec les heuristiques h_{cout}, h^+_{deg} et h_{freq}. Pour tout $i \in \{1,\ldots,14\}$ G_i est une sous-carte partielle de G_0 obtenue en supprimant $i.5\%$ de ses brins ou de ses coutures. Les trois courbes représentent respectivement l'évolution de $d^p_{1,1}$ et les approximations de $d^p_{1,1}$ et $d^i_{1,1}$ calculées. Les courbes sont des moyennes calculées sur un ensemble de 60 $2G$-cartes.*

cace de spécialiser les heuristiques en fonction des objets modélisés. Notez aussi que l'algorithme, avant de calculer la distance, effectue une mise en correspondance des brins. Cette mise en correspondance est utilisée pour calculer la distance d'édition. Il est donc tout à fait possible d'utiliser cette construction pour associer des parties similaires d'objets ou déterminer l'ensemble d'opérations permettant de modifier un objet en un autre sous forme de nG-carte.

Chapitre 6

Évaluation expérimentale

Sommaire

6.1 Base d'images 78
6.1.1 Modélisation d'images par des 2G-cartes et des graphes 78
6.1.2 Définition des étiquettes et des coûts 80
6.1.3 Description du jeu de données 84
6.1.4 Classification d'images 87
6.1.5 Comparaison des appariements construits 90
6.2 Base de maillages 95
6.2.1 Définition des étiquettes et coûts 100
6.2.2 Description des jeux de données 102
6.2.3 Classification de maillages 104
6.2.4 Comparaison des appariements construits 106
6.3 Discussion 110

Dans le Chap. 4 nous avons défini une distance entre nG-cartes basée sur la taille de la plus grande sous-carte commune. Nous avons ensuite défini la distance d'édition entre nG-cartes et montré que nous pouvons calculer sa valeur (sous certains coûts) à partir de la plus grande sous-carte commune. Dans le Chap. 5 nous avons proposé un algorithme calculant une approximation de la plus grande sous-carte commune en construisant un appariement de brins qui met en relation des brins de la première nG-carte avec ceux de la seconde.

Nous avons cherché une application directe de notre distance pour classer, reconnaître ou encore comparer des images ou des maillages $2D$ ou $3D$. Dans aucune de ces applications nous n'avons été réellement compétitifs face aux méthodes dédiées telles que, par exemple, [VNU03] une méthode basée sur des sac de mots pour la classification d'images, et [LSG+12] une méthode de classification de maillages en utilisant leur géométrie.

Les nG-cartes modélisent la structure d'un objet, et les distances que nous avons définies utilisent cette structure non seulement pour comparer les objets mais aussi pour mettre en correspondance leur composants et identifier leurs points communs et différences. En ce sens, notre approche s'insère dans le courant de la reconnaissance de formes structurelle (par opposition à la reconnaissance de formes statistiques). Par conséquent, nous allons nous comparer dans ce chapitre à la distance d'édition de graphes qui est l'approche la plus utilisée en reconnaissance de formes structurelles.

Notre objectif est notamment d'évaluer l'apport des cartes généralisées par rapport aux graphes pour faire de la reconnaissance de formes structurelles. Nous utilisons tout d'abord un jeu de 2G-cartes modélisant des images en niveau de gris segmentées et ensuite des 2G-cartes modélisant des maillages $3D$. Nous terminerons sur une discussion bilan sur les mesures de similarité basées sur des méthodes structurelles.

6.1 Base d'images

Les heuristiques proposées permettent de calculer de bons candidats à la plus grande sous-carte commune dans le cas de nG-cartes générées aléatoirement. Cependant la littérature montre que, dans la cadre de la distance d'édition de graphes, il est nécessaire pour comparer efficacement des objets d'introduire de l'information propre aux objets sous forme d'étiquettes. Par exemple, nous pouvons ajouter des étiquettes colorimétriques pour des images.

Nous allons valider ici $ApproxD$ sur une base d'images et nous commençons par décrire notre méthode de modélisation d'une image avec une 2G-carte. Nous introduirons ensuite un ensemble d'étiquettes permettant de décrire des informations colorimétriques et surfaciques de nos images segmentées. Enfin nous évaluerons les résultats obtenus en nous comparant à la distance d'édition de graphes pour effectuer de la classification ou de la mise en correspondance de zones de l'image.

6.1.1 Modélisation d'images par des 2G-cartes et des graphes

Pour mesurer la similarité entre des images, nous avons besoin de modéliser les images avec des structures. Nous allons proposer et comparer une modélisation à base de cartes généralisées ainsi qu'une modélisation à base de graphes.

Pour modéliser une image en utilisant des 2G-cartes, nous pouvons représenter chaque pixel de l'image par une face. Le défaut de cette méthode est que, sans étiquette, aucune information n'est intégrée à la structure de la 2G-carte autre que leur résolution. Deux images de même résolution seront donc modélisées par des 2G-cartes isomorphes. Afin d'étoffer l'information contenue dans la structure de la 2G-carte, nous appliquons un procédé de segmentation sur l'image qui permet d'associer à chaque face un ensemble de pixels connexes et similaires au lieu d'un unique pixel. Pour segmenter les images nous utilisons l'algorithme de Damiand et Dupas décrit dans [DD08]. Cet algorithme construit tout d'abord la 2G-carte modélisant l'image avec une face par pixel. Dans une première étape, il fusionne les faces adjacentes ayant un gradient de couleur faible qui indique que les couleurs des deux faces sont proches. Dans un second temps, il fusionne les arêtes afin qu'entre deux faces voisines, il n'y ait qu'une unique arête. Le résultat de ce processus est une image segmentée. La Fig. 6.1 illustre la segmentation d'une image de 24 pixels, en trois régions avec cet algorithme. Elle montre aussi une image ainsi que la modélisation de sa segmentation avec une 2G-carte.

Afin de comparer notre distance d'édition à la distance d'édition des graphes nous modélisons également chaque image par un graphe. Afin que l'algorithme de segmentation utilisé n'influe pas sur les résultats, nous utilisons le même algorithme que pour les 2G-cartes.

La structure de graphe la plus connue pour modéliser des images segmentées est le graphe d'adjacence des régions (RAG). Ce graphe associe un sommet à chaque région de l'image et possède une arête entre deux sommets si les deux régions associées ont au moins un pixel adjacent. La Fig. 6.1 illustre une image, sa segmentation ainsi que le

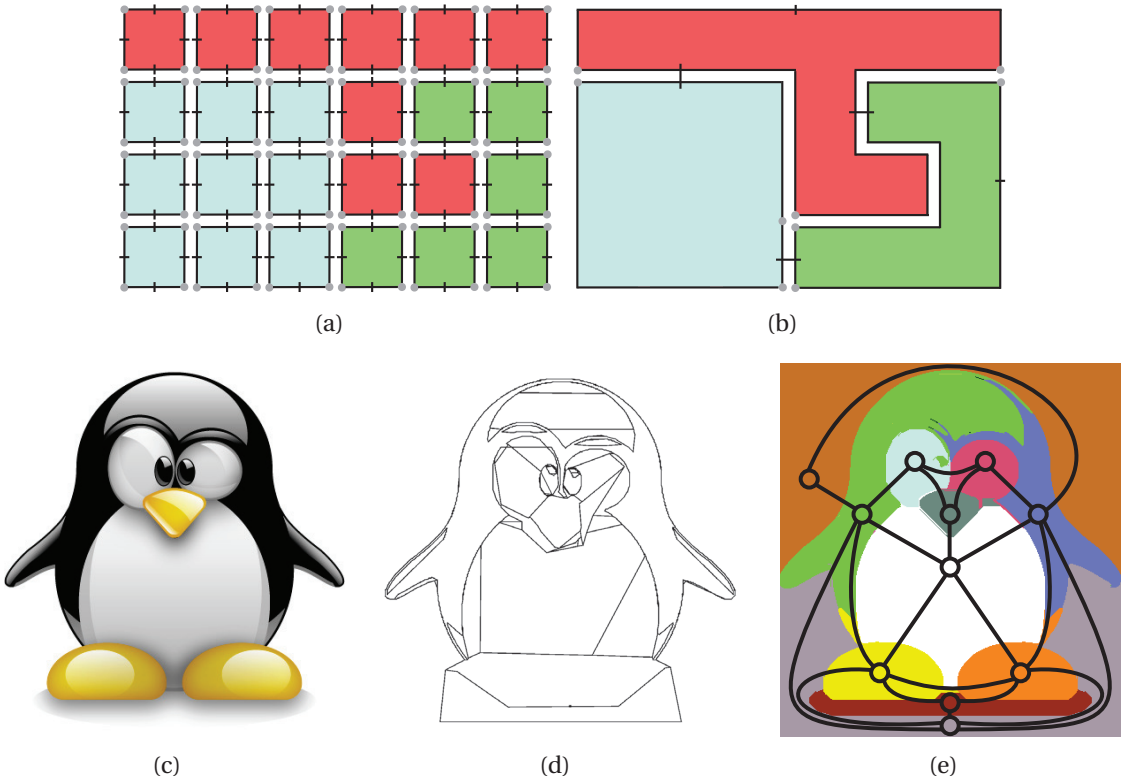


FIGURE 6.1 – (a) *Exemple d'une modélisation d'une image avec une* 2*G-carte (une face par pixel).* (b) *La* 2*G-carte obtenue après segmentation par [DD08].* (c) *Une image.* (d) *Sa modélisation avec une* 2*G-carte.* (e) *Sa modélisation avec un graphe d'adjacence de région.*

graphe d'adjacence de région la modélisant. Cependant, certaines informations ne sont pas disponibles avec le graphe d'adjacence de régions, notamment le nombre d'arêtes composant les régions alors qu'elles sont disponibles dans la structure des 2G-cartes. Dans un souci d'équité nous ajoutons des étiquettes supplémentaires sur le graphe :

- une étiquette sur chaque sommet du graphe indiquant le nombre d'arêtes qui composent la face de l'image segmentée ;
- une étiquette sur chaque arête du graphe indiquant le nombre d'arêtes de l'image composant la jointure entre deux régions.

La Fig. 6.2 donne un exemple de graphe d'adjacence de régions étiqueté. Notez cependant que deux images peuvent être représentées avec le même graphe, alors qu'elles seront représentées par deux 2G-cartes non isomorphes. Par exemple dans la Fig. 6.2, les images (a) et (b) sont modélisées par les graphes (c) et (d) qui sont isomorphes alors qu'elles sont modélisées par les 2G-cartes (e) et (f) qui ne sont pas isomorphes. Cette différence est inhérente au fait que les 2G-cartes portent une information de plongement dans un plan qui n'est pas intégrée aux graphes. Ainsi, toutes les images correspondant à un plongement différent d'un même graphe seront représentées par un graphe unique. Cette différence nous permet d'obtenir des résultats différents.

6.1.2 Définition des étiquettes et des coûts

Nous proposons également d'utiliser des étiquettes pour le cas des images, l'une basée sur la colorimétrie et la seconde basée sur la surface des régions issues de la segmentation.

Couleur

L'algorithme de segmentation produit des régions dont les pixels sont de couleur similaire. Il associe à chacune de ces régions leur couleur moyenne. Cette information peut être ajoutée comme étiquette sur les 2G-cartes comme sur les graphes.

Lorsque l'image est modélisée par un RAG, la couleur de chaque région est la couleur moyenne de tous les pixels composant la région correspondante. Nous ajoutons donc une étiquette à chaque sommet contenant la couleur de la région qu'il représente.

Dans une image modélisée par une 2G-carte, nous proposons de déplacer cette information de la face (qui est décrite implicitement avec les orbites) vers ses brins. Chaque brin appartenant exactement à une face, nous associons à chaque brin une étiquette contenant la couleur de sa face.

Dans les expérimentations menées ici, les couleurs sont des niveaux de gris et chaque étiquette est donc une valeur entière comprise entre 0 et 255. Si nous souhaitons représenter des étiquettes de couleurs (RGB), il conviendra d'utiliser une étiquette qui permettra de faciliter le calcul de distance entre couleurs. Pour cela, [CIE94, SWD] proposent par exemple des systèmes de couleurs adaptés, et des distances qui correspondent à la perception de l'oeil humain.

Nous définissons le coût de substitution de l'étiquette l en l', pour les 2G-cartes comme pour les graphes :

$$c_{subst-col}(l,l') = \frac{|l-l'|}{255} \tag{6.1}$$

Ce coût est borné entre 0 et 1. Il est égal à 0 si les couleurs des sommets ou des brins sont identiques et à 1 si les couleurs sont les plus éloignées (0 et 255). Cette première

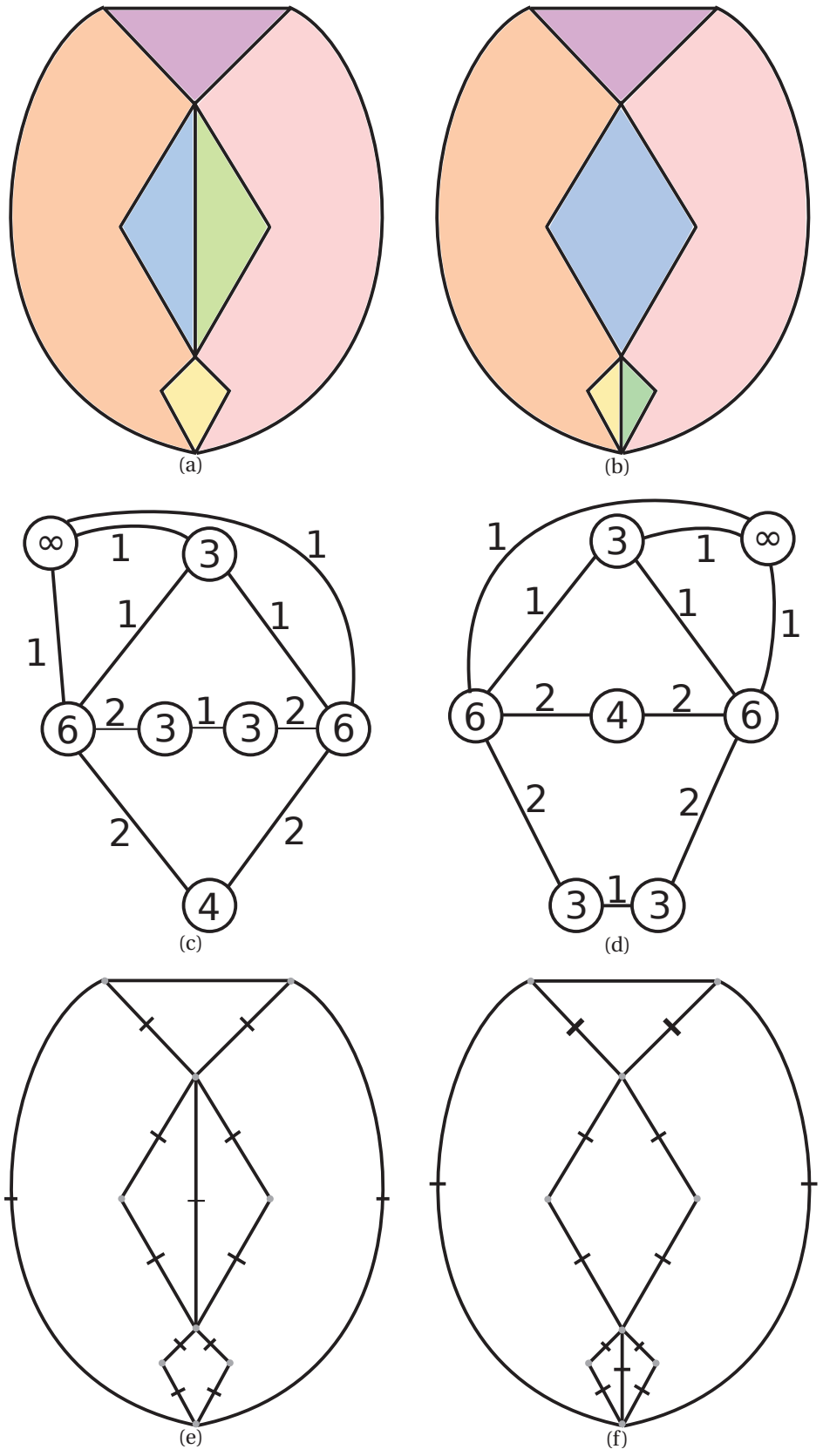


FIGURE 6.2 – *(a) et (b) deux images, (c) et (d) leur modélisation en utilisant un graphe d'adjacence de région. (e) et (f) leur modélisation en utilisant une 2G-carte. Les graphes (c) et (d) sont isomorphes tandis que les 2G-cartes (e) et (f) ne sont pas isomorphes. Notez que la somme du coût des arêtes est égal au coût des sommets.*

étiquette et son coût de substitution va permettre à la distance d'exprimer que deux brins ou deux sommets appariés sont d'autant plus éloignés que leurs couleurs sont éloignées.

Aire des faces

La seconde étiquette que nous pouvons ajouter est liée à la taille des régions issues de la segmentation. En effet, suite à l'aggrégation des pixels de couleurs similaires dans le processus de segmentation, les régions de l'image segmentée sont plus ou moins grandes. Une image étant composée exclusivement de pixels, nous pouvons calculer l'aire d'une région en comptant le nombre de pixels qui la composent. Ainsi, l'étiquette proposée a pour valeur le nombre de pixels aggregés dans la région.

Pour les graphes, les sommets représentant les régions, nous ajoutons une étiquette sur chaque sommet. Pour les 2G-cartes, cette information étant encore une fois propre à la face, nous déportons l'étiquette sur chacun des brins. Ainsi, sur chaque brin, nous ajoutons l'étiquette correspondant à la l'aire de la face qui lui est incidente.

Dans les deux cas, le coût de substitution proposé est défini en fonction de la différence de l'aire des régions. Ainsi, le coût de substitution de l'étiquette l en l'étiquette l' respectivement d'aires A et A' est égal à :

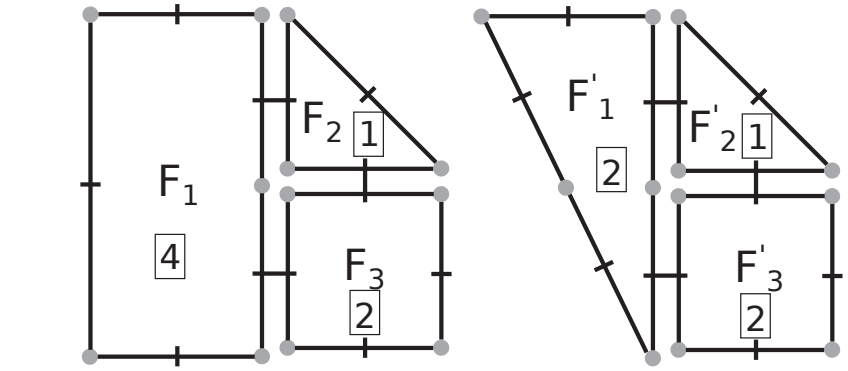
$$c_{subst-aire}(l,l') = 1 - \frac{min(A,A')}{max(A,A')} \tag{6.2}$$

Ce coût de substitution est borné entre 0 et 1 : égal à 0 si les deux aires sont égales et d'autant plus proche de 1 que le ratio de l'aire entre les deux faces augmente.

Dans la Fig. 6.3, nous utilisons l'étiquette basée sur l'aire des faces. La distance d'édition entre G et G' est égal à 20, ce qui correspond au coût de substitution des 10 brins de la face F_1 avec les 10 brins de la faces F'_1. En effet, les brins de F_1 étant étiquetés avec une aire de 4 et ceux de F'_1 avec une aire de 2, le coût de substitution de chaque couple est donc de 2. Notez que pour cette étiquette dans le cas des 2G-cartes, le coût de substitution est proportionnel au nombre de brins composant les faces, c'est différent pour les graphes où la distance d'édition correspond au coût de substitution du sommet représentant la face F_1 avec le sommet représentant la face F_2, i.e., 2 dans l'exemple de la Fig. 6.3.

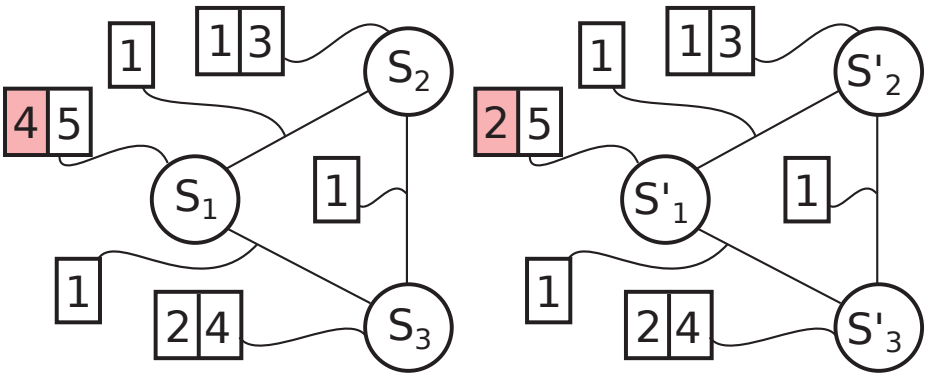
Définition des coûts d'édition

Pour ces expérimentation nous avons choisi des coûts d'éditions pour chacune des opérations. Pour les graphes nous avons fixé le coût d'ajout d'un sommet (que nous noterons $c(add_n)$) et de suppression d'un sommet (que nous noterons $c(del_n)$) à 1. De même pour les opérations ajoutant et supprimant une arête, i.e., $c(add_e) = c(del_e) = 1$. Le coût de substitution est différent en fonction des étiquettes. Pour les étiquettes indiquant la couleur, nous choisissons $c_{subst-col}$ et pour les étiquettes indiquant l'aire des régions $c_{subst-aire}$. Pour les étiquettes structurelles, c'est-à-dire l'étiquette sur les arêtes indiquant le nombre de fois que deux faces sont adjacentes et l'étiquette sur les sommets indiquant le nombre d'arêtes composant la face nous avons choisi une fonction de coût renvoyant la différence entre les deux étiquettes, que nous noterons $c(subst-struct(l,l')) = |l-l'|$. Enfin, le coût de substitution final entre deux sommets est égal à la somme des coûts de substitutions. Ainsi le coût de substitution de deux sommets s par s' est égal à la somme des trois coûts de substitution $c(subst - struct(l,l'))$, $c_{subst-col}$ et $c_{subst-aire}$ en fonction des étiquettes considérées dans la distance d'édition.



(a) G (b) G'

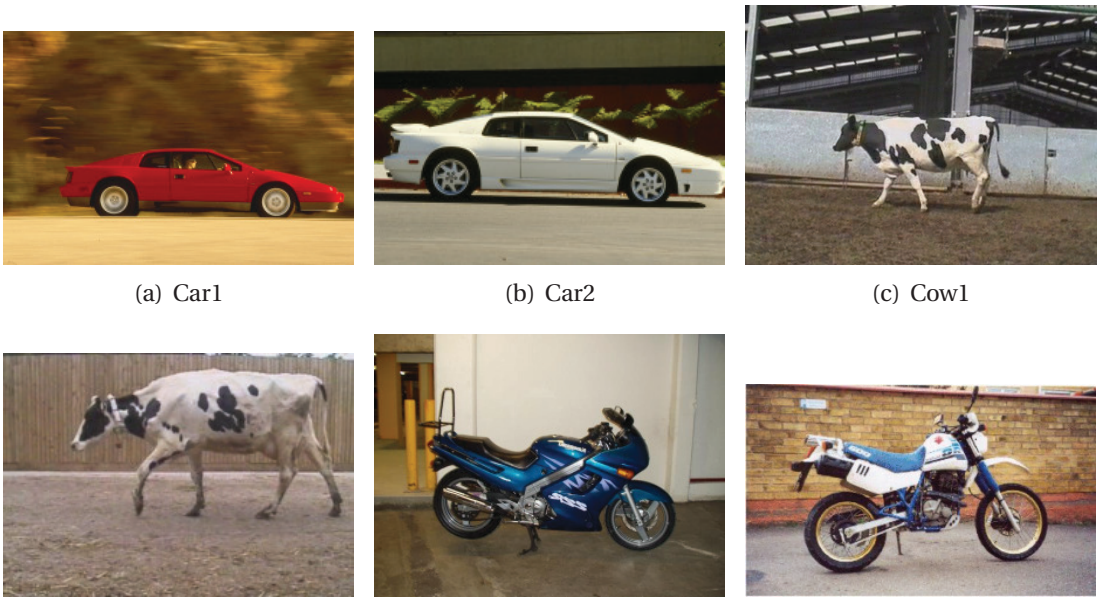
Coût associé à l'appariement $\{(F_1, F'_1), (F_2, F'_2), (F_3, F'_3)\} = 10 \times 2 = 20$



(c) g (d) g'

Coût associé à l'appariement $\{(S_1, S'_1), (S_2, S'_2), (S_3, S'_3)\} = 2$

FIGURE 6.3 – *Exemple de distances d'édition sur les graphes et les 2G-cartes où seul le coût de substitution de l'aire des faces est pris en compte. (a) et (b) sont deux 2G-cartes modélisant une image composée de 3 faces. (c) et (d) sont deux graphes modélisant les mêmes images, les sommets S_1, S_2, S_3 (resp. S'_1, S'_2, S'_3) représentant les faces F_1, F_2, F_3 (resp. F'_1, F'_2, F'_3). Les étiquettes sur les arêtes sont les étiquettes indiquant le nombre de relation d'adjacence entre les deux faces. Les étiquettes sur les sommets sont : l'aire pour l'étiquette de gauche, et le nombre d'arête composant la face qu'il représente à droite.*



(a) Car1 (b) Car2 (c) Cow1

(d) Cow2 (e) Moto1 (f) Moto2

FIGURE 6.4 – *Exemples d'images extraites de la base ETHZ.*

Pour les 2G-cartes, nous utilisons les coûts proposés $c^p_{1,1}$ dans la Section 4.5. Pour rappel nous avons donc, pour deux cartes comparées G et G' :

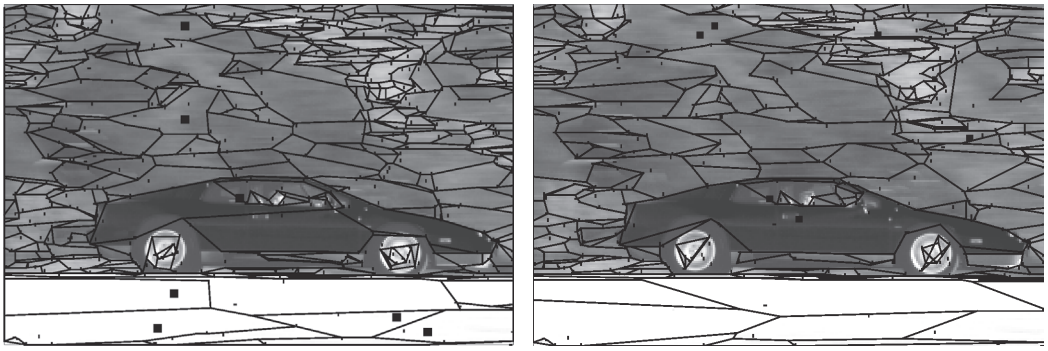
- $c^p_{1,1}(del_E) = |E|\,|\{(d_1, i, d_2) \in coutures_{G'}(D'), d_i \in E \text{ où } d_2 \in E\}|$;
- $c^p_{1,1}(add_{E,F}) = |E| + |F|$;
- $c^p_{1,1}(sew_F) = c^p_{1,1}(unsew_F) = |F|$.

Pour le coût de substitution, nous considérons une somme des coûts de substitution des étiquettes $c_{subst-aire}$ et $c_{subst-col}$.

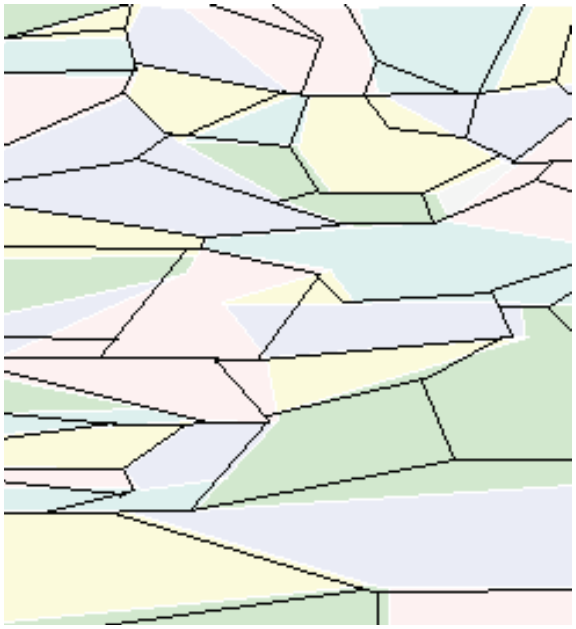
6.1.3 Description du jeu de données

Notre objectif est de comparer différentes segmentations d'une même image. Comparer deux segmentations d'une même image nous permet d'avoir une "vérité terrain" sur les zones mises en correspondance. Nous pourrons ainsi comparer la qualité des appariements construits pour calculer les distances ce qui n'est pas possible quand nous comparons deux images différentes pour lesquelles nous ne savons quelle sont les zones qui doivent être appariées. Nous sommes partis des 6 images de la Fig. 6.4 extraites de la base ETHZ [1]. Pour chaque image, nous avons généré différentes segmentations en utilisant l'algorithme de segmentation de [DD08] avec différentes valeurs de seuils. La Fig. 6.5 donne un exemple de deux segmentations de la même image. La Fig. 6.5(a) est une segmentation plus fine que celle de la Fig. 6.5(b). Notez que toutes les segmentations sont recalculées à partir de l'image initiale. Ainsi, les régions d'une segmentation plus grossière ne sont pas forcément des aggrégations des régions de la segmentation plus fine. Par exemple, sur la Fig. 6.5(c) nous avons effectué un zoom de l'image, représenté en couleur les régions de la segmentation de la Fig. 6.5(a), et par des traits noirs la segmentation de la Fig. 6.5(b). Notez que certaines faces coloriées chevauchant plusieurs régions de la Fig. 6.5(b) et inversement.

1. disponible sur http ://pascallin.ecs.soton.ac.uk/challenges/VOC/databases.html



(a) Nombre de brins : 4774 (b) Nombre de brins : 4060



(c) Comparaison de la segmentation

FIGURE 6.5 – *(a) et (b) Exemple de deux segmentations successives d'une image. (c) Un agrandissement des deux segmentations, la première est représentée par des traits noirs, la seconde par des couleurs.*

Classe	#cartes	#brins
		moy(min max)
			#coutures
			moy(min max)
				#sommets
				moy(min max)
					#arêtes
					moy(min max)
						#faces
						moy(min max)

Car1	10	2952(4776 3785)	11355(8856 14328)	625(487 787)	946(738 1194)	336(263 421)
Car2	7	2792(4548 3701)	11105(8376 13644)	614(463 753)	925(698 1137)	317(243 390)
Cow1	15	2992(4776 3870)	11612(8976 14328)	646(504 792)	967(748 1194)	371(298 458)
Cow2	13	2932(4938 3887)	11663(8796 14844)	641(484 814)	971(733 1237)	337(255 431)
Moto1	19	2872(4724 3694)	11084(7616 14172)	611(475 783)	923(718 1181)	317(251 406)
Moto2	21	2952(4932 3838)	11514(8856 14796)	635(489 816)	959(738 1233)	335(265 425)

Tableau 6.1 – *Description des 2G-cartes utilisées dans l'expérimentation.*

Classe	#sommets
	moy(min max)
		#arêtes
		moy(min max)

Car1	336(263 421)	922(717 1171)
Car2	317(243 390)	911(688 1124)
Cow1	371(298 458)	918(697 1146)
Cow2	337(255 431)	950(704 1217)
Moto1	317(251 406)	904(702 1159)
Moto2	335(265 425)	934(710 1209)

Tableau 6.2 – *Description des graphes utilisés dans l'expérimentation.*

Le jeu de données utilisé est donc composé d'un ensemble de segmentations des 6 images de Fig. 6.4. Nous souhaitons ainsi utiliser l'ensemble des segmentations issues d'une image pour définir une classe. Pour générer nos classes nous avons fait varier incrémentalement le seuil de l'algorithme de segmentation et conservé toutes les images ayant un nombre de régions compris entre 240 et 460.

Le Tableau 6.1 décrit les 2G-cartes modélisant les images issues de la segmentation, le Tableau 6.2 décrit les graphes construits à partir des images segmentées et enfin le Tableau 6.3 décrit l'ensemble des étiquettes associées aux régions des images segmentées (qui seront donc associées aux sommets des graphes et aux brins des 2G-cartes).

Dans ces expérimentations, nous allons aussi comparer les images en fonction du nombre de segmentations qui les séparent. Nous détaillons dans le Tableau 6.4 la différence moyenne en nombre de brins/sommets entre deux segmentations de niveaux différents. Par exemple pour deux images séparées par deux segmentations, leur différence moyenne en nombre de brins sera de 277 pour les 2G-cartes, alors que pour les graphes, elle sera de 23 sommets.

Notons que, malgré le fait que les 6 images originales soient différentes, la segmentation produit des cartes structurellement très similaires. En effet, leurs nombres de sommets, arêtes, faces, brins ou coutures sont très proches indépendemment de leur classe. Par contre, au niveau des étiquettes, on perçoit de vraies différences autant au niveau des couleurs que de l'aire moyenne des régions segmentées. Ainsi avec notre jeu

Classe	couleur moy(min max)	aire moy(min max)
Car1	87(0 214)	193(1 6019)
Car2	119(0 244)	122(1 4450)
Cow1	111(0 215)	980(1 19510)
Cow2	125(0 229)	134(1 3400)
Moto1	88(0 228)	230(1 9890)
Moto2	120(0 243)	173(1 5004)

Tableau 6.3 – *Statistique sur les régions des images segmentées.*

S	#brins	#sommets
0	0	0
1	143	12
2	277	23
3	409	34
4	536	44
5	657	54
6	755	63
7	844	70
8	942	79
9	1035	86
10	1118	92

Tableau 6.4 – *Différence moyenne en nombre de brins (#b) et sommets (#s) entre S niveaux de segmentations. Par exemple, entre le niveau de segmentation i, et le niveau de segmentation i+5 les cartes diffèrent en moyenne de 657 brins et les graphes de 54 sommets.*

d'essai, un simple histogramme de couleur est capable d'avoir un taux de bonne classification de 100% (puisque deux même images d'une même classe ont sensiblement le même histogramme de couleurs). En revanche, un histogramme de couleurs ne permettra pas de retrouver, dans deux images d'une même classe, quelles sont les zones correspondantes (au delà des zones ayant la même couleur). Nous souhaitons évaluer ici la capacité des approches structurelles à apparier correctement les zones de deux segmentations issues d'une même image. L'avantage d'utiliser deux segmentations différente d'une même image (plutôt que des segmentations de deux images différentes) est que nous disposons d'une "vérité terrain" pour décider si deux zones sont correctement appariées ou non.

6.1.4 Classification d'images

Pour les 2G-cartes comme pour les graphes, nous avons donc un ensemble de 85 éléments à classer dans 6 classes correspondant aux 6 images initiales. Pour la classification, nous comparons tous les couples d'objets et nous utilisons l'algorithme 'k nearest neighbors' [SJ89] (k-nn) classique pour effectuer la classification. Nous utilisons le procédé du 'leave one out' pour lequel nous essayons de classer chaque élément en connaissant les classes de tous les autres. Nous procédons avec un système de vote en considérant pour chaque vote les k éléments ayant la distance la plus petite. Pour cette expérimentation, pour les 2G-cartes, nous lançons 10 exécutions de l'algorithme $ApproxD$ avec les heuristiques h^+_{deg}, h_{dyn} et h^+_{freq} et considérons pour chaque heuristique, la meilleure approximation calculée sur les 10 exécutions. Pour les graphes nous utilisons la distance d'édition décrite dans [RB09] et rappelée en Section 2.3. Dans les deux cas, nous faisons varier les étiquettes en ne considérant dans un premier temps aucune étiquette, puis dans un second temps en considérant l'étiquette indiquant l'aire des régions, ensuite avec la colorimétrie et enfin avec la combinaison des deux. Notez que pour les graphes, nous conservons toujours les étiquettes apportant des informations structurelles sur les sommets et arêtes (les résultats sont nettement moins bons lorsque ces étiquettes structurelles ne sont pas prises en compte). Enfin nous implémentons aussi une méthode naïve calculant une simple différence entre les histogrammes de couleur des images segmentées. Ainsi, pour l'histogramme de couleur, la distance entre deux images est simplement la somme des différences du nombre de ré-

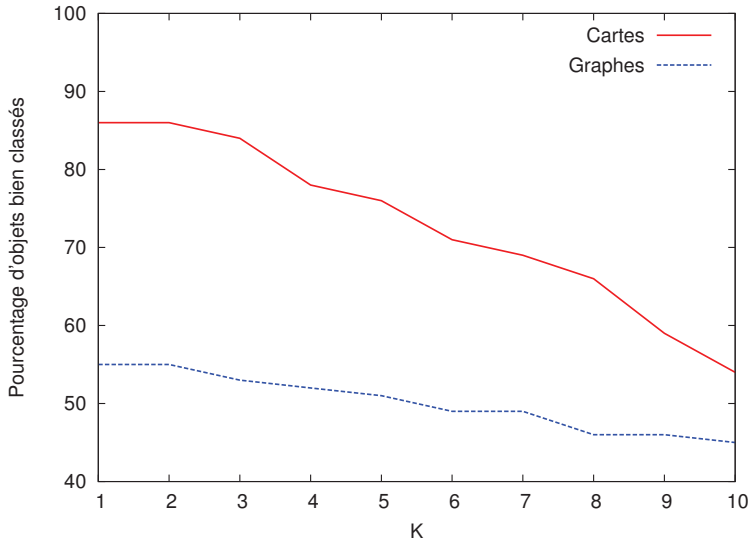


FIGURE 6.6 – *Résultats de classification des images en faisant varier le paramètre k de l'algorithme du k-nn. La courbe "Graphes" correspond aux résultats obtenus par l'algorithme [RB09] sans étiquettes. La courbe "Cartes" correspond aux résultats obtenus par les 2G-cartes.*

gions ayant la même couleur moyenne. Cette méthode permet de mesurer la capacité discriminante de la colorimétrie pour la classification de notre jeu de données.

Résultats

Comparons tout d'abord les résultats de classification en ne considérant que l'information structurelle de sorte que les 2G-cartes ne sont pas étiquetées. Les graphes ont cependant toujours les étiquettes introduites dans la Section 6.1.2 décrivant la structure (nombre d'arêtes par région, et nombre d'adjacences entre régions). Ensuite nous étudierons l'impact des étiquettes sur les taux de bonne classification.

Dans la Fig. 6.6, nous comparons les résultats de la classification en fonction de la valeur de k de l'algorithme du k-nn. Cette expérimentation permet d'évaluer la capacité de classification de segmentations d'images en utilisant uniquement des informations structurelles. Notez que l'algorithme [RB09] n'arrive à classer correctement que 55% des images, alors que notre distance permet un taux de bonne classification de plus de 86% des 2G-cartes.

Ensuite nous proposons d'étudier les résultats en ne considérant que l'étiquette basée sur l'aire des régions. Dans la Fig. 6.7 nous comparons les résultats de la classification. Notez que l'ajout de l'étiquette 'Aire' améliore le taux de bonne classification que ce soit pour les graphes ou pour notre algorithme. Cette fois, pour de petites valeurs de k, les deux méthodes permettent de classifier 83 images sur les 85. Quand la valeur de k augmente, les 2G-cartes permettent de classer correctement un peu plus d'images que les graphes, ce qui traduit une plus grande stabilité. En effet, cela implique que la distance est plus petite pour les autres segmentations issues de la même image que les segmentations issues d'autres images, même si leur nombre de sommets/brins diffèrent.

Nous ne considérons maintenant que l'étiquette colorimétrique. La Fig. 6.8 illustre les résultats de cette classification. La classification à base d'histogramme de couleur permet de classer correctement 100% des images quelle que soit la valeur de k.

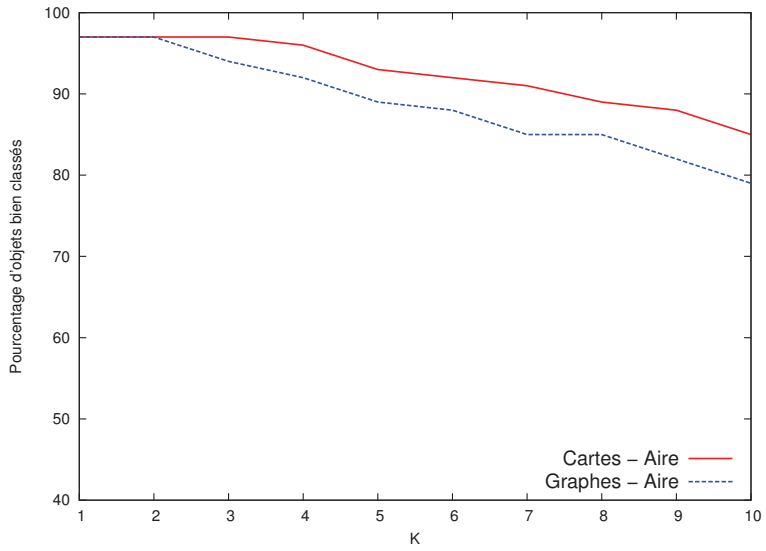


FIGURE 6.7 – *Résultats de classification des images en faisant varier le paramètre k de l'algorithme du k-nn. La courbe "Graphes" correspond aux résultats obtenus par l'algorithme [RB09] avec l'étiquette indiquant l'aire des régions. La courbe "Cartes" correspond aux résultats obtenus par les $2G$-cartes avec l'étiquette portant l'aire des régions.*

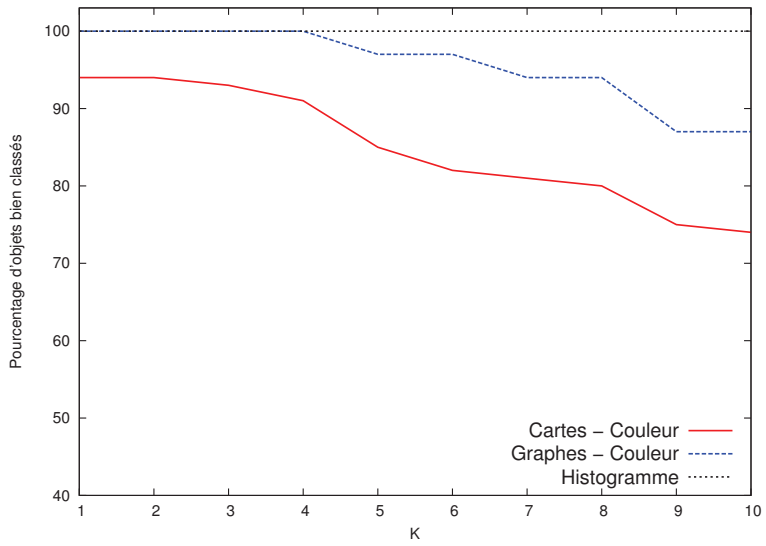


FIGURE 6.8 – *Résultats de classification des images en faisant varier le paramètre k de l'algorithme du k-nn. La courbe "Graphes" correspond aux résultats obtenus par l'algorithme [RB09] avec l'étiquette indiquant la couleur moyenne des régions. La courbe "Cartes" correspond aux résultats obtenus par les $2G$-cartes avec l'étiquette portant sur la couleur moyenne des régions. La courbe Histogramme effectue une classification simple en utilisant une distance basée sur une comparaison d'histogrammes de couleurs des images.*

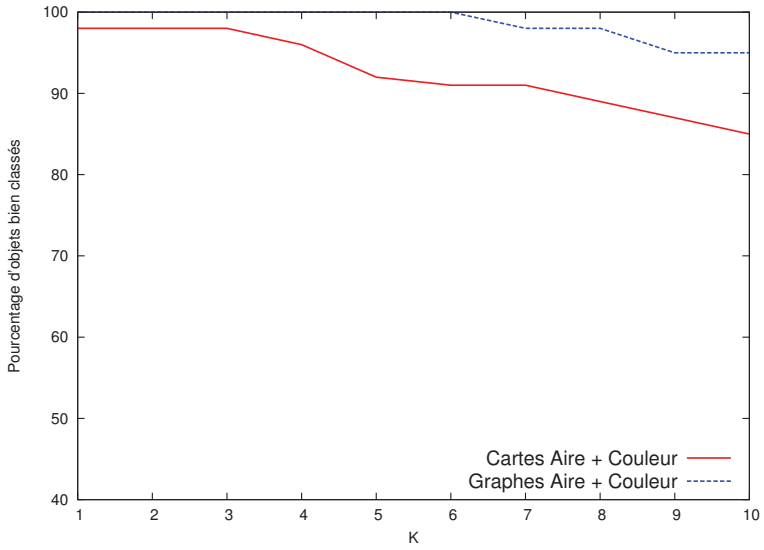


FIGURE 6.9 – *Résultats de classification des images en faisant varier le paramètre k de l'algorithme du k-nn. La courbe "Graphes" correspond aux résultats obtenus par l'algorithme [RB09] avec les étiquettes décrivant l'aire des régions et la couleur moyenne des régions. La courbe "Cartes" correspond aux résultats obtenus par les 2G-cartes avec les deux étiquettes.*

Nous mesurons donc dans cette expériementation la capacité des deux algorithmes à prendre en compte les étiquettes. La méthode utilisant les graphes classe cinq images de plus que les 2G-cartes pour les petites valeurs de k. Pour les grands valeurs de k, les graphes sont meilleurs que les 2G-cartes. Nous en concluons que les graphes utilisent mieux les étiquettes que les 2G-cartes. Cela est principalement du à notre algorithme. En effet, il se sert de la colorimétrie pour choisir un premier couple de départ, mais c'est la structure qui guide la construction dans un second temps. Pour les graphes la structure est peu utilisée, en effet, le graphe n'est utilisé que pour représenter la relation de voisinage entre sommets. La structure ayant peu d'impact, l'algorithme classifie les graphes en réalisant une mise en correspondance des étiquettes colorimétriques, qui sont particulièrement discriminantes.

Enfin, nous proposons d'étudier les taux de bonne classification quand nous utilisons une combinaison des deux étiquettes 'Aire' et 'Couleur'.

La Fig. 6.9 illustre les résultats de cette classification. Il est clair que la combinaison des étiquettes permet l'amélioration du taux de bonne classification avec les deux algorithmes.

6.1.5 Comparaison des appariements construits

L'algorithme [RB09] permet la construction d'un appariement de graphes à tolérance d'erreur appariant ses sommets. Notre algorithme construit un appariement de brins pour calculer la distance entre deux nG-cartes. Dans cette expérimentation nous allons comparer les appariements construits pour vérifier que les composants (sommets ou brins) mis en relation sont effectivement associés à des régions proches dans les deux images comparées. Pour les graphes, nous considérerons que deux *sommets sont bien appariés* si les deux régions correspondantes ont au moins 1 pixel en commun. Pour les 2G-cartes, nous considérerons que deux *brins sont bien appariés* si leurs faces incidentes (représentant les deux régions) ont au moins un pixel en commun.

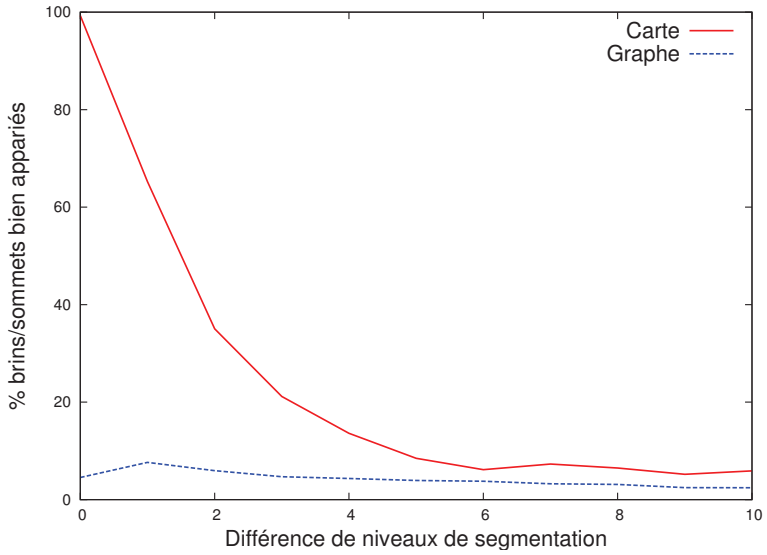


FIGURE 6.10 – *Taux de brins/sommets apparié de façon cohérente par rapport au nombre de segmentation existant entre les deux images segmentées comparées. On compare la segmentation i et la segmentation $i+j$, j étant représenté en abscisse.*

Nous utilisons ici, le même jeu de donnée que l'expérimentation précédente, décrit dans la Section 6.1.3

Nous avons traité chaque classe séparément en comparant tous les éléments entre eux. Pendant cette comparaison nous avons extrait les appariements construits par notre algorithme et celui de [RB09]. Puis nous avons noté le pourcentage de brins et sommets bien appariés par rapport au nombre de niveaux de segmentation qui séparent les images comparées. Nous avons utilisé le même ensemble de segmentations que pour l'expérimentation de classification. La Fig. 6.10 détaille le nombre de brins et sommets qui diffèrent en moyenne en fonction du nombre de niveaux de segmentation séparant les images comparées.

Résultats

Dans un premier temps, nous comparerons le pourcentage de brins et de sommets bien appariés en ne considérant ni l'étiquette 'Aire', ni l'étiquette 'Couleur'. La Fig. 6.10 montre que l'algorithme utilisant les graphes ne permet pas d'apparier des sommets de façon cohérente, alors que notre algorithme calcule des appariements permettant d'associer des brins de façon cohérente. Étudions par exemple le cas où la différence du niveau de segmentation est de 0, c'est-à-dire que nous comparons des 2G-cartes et graphes isomorphes. Pour les graphes, l'appariement calculé par l'algorithme n'associe pas les sommets avec les bons sommets correspondants. Cela est du au fait que l'algorithme ne considère les choix localement. En revanche, pour les 2G-cartes, l'algorithme apparie bien l'ensemble des brins dans le cas où nous comparons la même image. Si nous augmentons le nombre de segmentations entre les images comparées, notre algorithme ne retrouve plus que 40% des brins pour une différence de deux niveaux de segmentation (soit environ 277 brins de différence en moyenne, donc entre 5% et 10% de différence en nombre de brins en fonction des images comparées).

Dans la Fig. 6.11 nous ajoutons l'étiquette 'Aire' aux graphes et aux 2G-cartes et notons une amélioration du nombre de sommets/brins effectivement bien appariés. Ce-

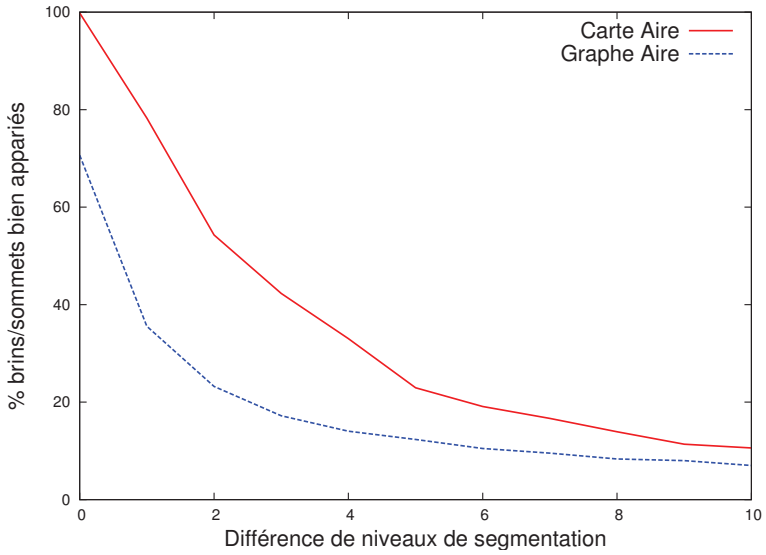


FIGURE 6.11 – *Taux de brins/sommets appariés de façon cohérente par rapport au nombre de segmentations existantes entre les deux images segmentées comparées. On compare la segmentation i et la segmentation $i+j$, j étant représentée en abscisse. On considère ici l'étiquette indiquant l'aire des régions sur les brins et sommets.*

pendant, notre algorithme construit un appariement qui apparie bien un plus grand pourcentage de brins. Les résultats sont similaires pour l'étiquette 'Couleur' dans la Fig. 6.12 et quand nous considérons une combinaison des deux étiquettes (Aire et Couleur) dans la Fig. 6.13.

Il est intéressant de mettre ce résultat en perspective avec le fait que la distance d'édition de graphes a de meilleurs taux de bonne classification que les cartes généralisées lorsqu'elle exploite les étiquettes de couleur. On explique ce résultat par le fait que les graphes n'utilisent que peu la structure. En effet, elle sert au moment de la création de la matrice pour définir le voisinage des sommets, ils effectuent donc une classification en comparant les étiquettes des deux graphes. Il est alors très simple d'associer des sommets de couleurs proches, sans pour autant qu'ils représentent la même zone.

Étudions maintenant les deux structures indépendamment pour évaluer l'apport de nos étiquettes sur la qualité de l'appariement. La Fig. 6.14 montre toutes les courbes correspondant aux résultats obtenus par les graphes. Nous notons que, bien que peu de sommets sont bien appariés lorsque nous ne considérons pas d'étiquettes, l'ajout d'une étiquette 'Aire' permet d'améliorer les résultats. La couleur semble être encore plus discriminante, et enfin la combinaison des deux permet d'obtenir les meilleurs taux de sommets bien appariés.

Pour les 2G-cartes, nous obtenons des résultats similaires illustrés dans la Fig. 6.15, avec un meilleur taux de brins bien appariés obtenu avec la combinaison des deux étiquettes, suivi de la couleur moyenne des régions, de l'aire et enfin en ne considérant aucune des deux étiquettes. Notez que contrairement aux graphes, l'algorithme $ApproxD$ permet de bien apparier la totalité des brins si nous considèrons une différence de niveau de segmentation de 0 (pour laquelle les 2G-cartes comparées sont isomorphes). Nous expliquons cette différence du fait que l'algorithme calculant un appariement permettant d'approximer la distance d'édition de graphes (décrit en Section 2.3) ne considère pas la structure du graphe de façon globale et n'est pas capable de trouver l'appariement correspondant à l'isomorphisme.

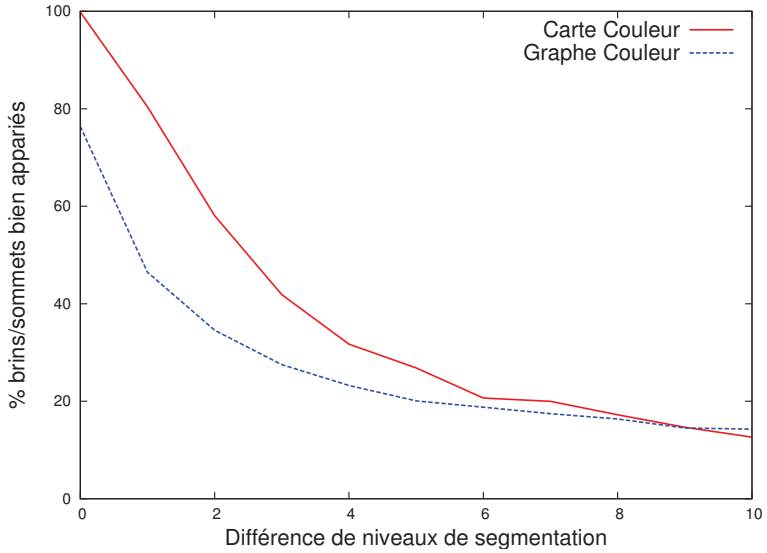


FIGURE 6.12 – *Taux de brins/sommets appariés de façon cohérente par rapport au nombre de segmentations existantes entre les deux images segmentées comparées. On compare la segmentation i et la segmentation $i+j$, j étant représentée en abscisse. On considère ici l'étiquette indiquant la couleur moyenne des régions sur les brins et sommets.*

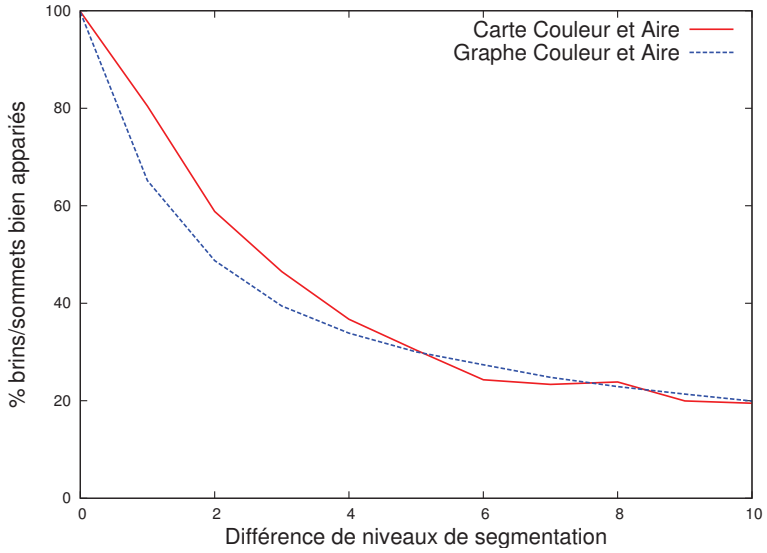


FIGURE 6.13 – *Taux de brins/sommets appariés de façon cohérente par rapport au nombre de segmentations existantes entre les deux images segmentées comparées. On compare la segmentation i et la segmentation $i+j$, j étant représentée en abscisse. On considère ici les étiquettes indiquant l'aire des régions et leur couleur moyenne sur les brins et sommets.*

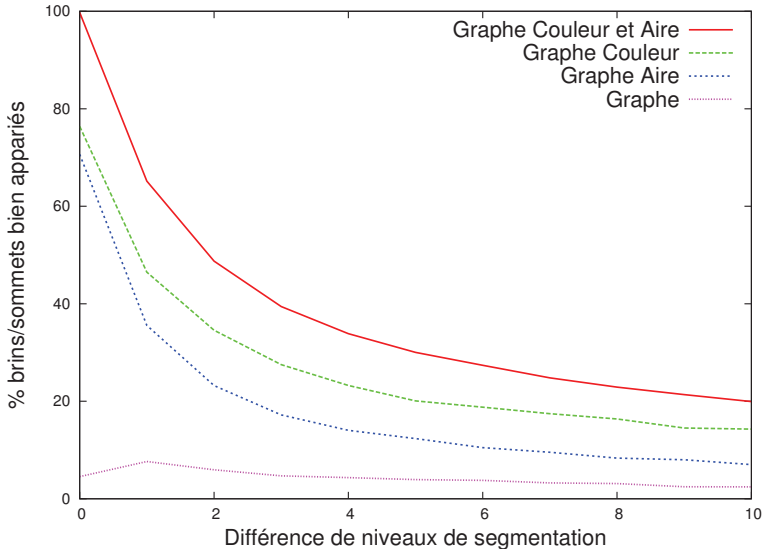


FIGURE 6.14 – *Taux de brins/sommets appariés de façon cohérente par rapport au nombre de segmentations existantes entre les deux images segmentées comparées. On compare la segmentation i et la segmentation $i+j$, j étant représentée en abscisse. Les courbes correspondent aux résultats obtenus en comparant les graphes. Nous affichons une courbe par type d'étiquette considéré.*

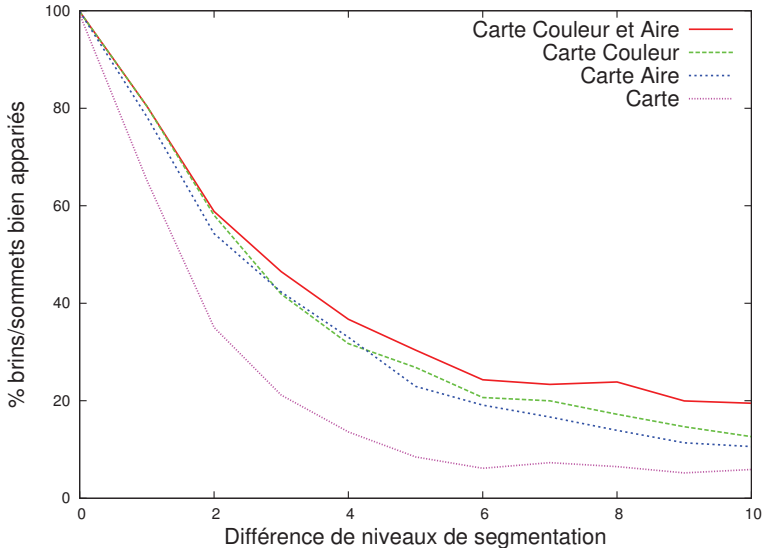


FIGURE 6.15 – *Taux de brins/sommets appariés de façon cohérente par rapport au nombre de segmentations existantes entre les deux images segmentées comparées. On compare la segmentation i et la segmentation $i+j$, j étant représentée en abscisse. Les courbes correspondent aux résultats obtenus en comparant les $2G$-cartes. Nous affichons une courbe par type d'étiquette considéré.*

Affichage des résultats

Chaque brin est incident à une face, et donc appartient à une région. Nous pouvons donc associer tous les pixels de la région à ce brin. Pour afficher notre résultat nous créerons une image composée des pixels appartenant à l'intersection des faces dont au moins un de leurs brins est apparié. Par exemple, si un brin b est apparié au brin b' alors nous conserverons l'ensemble des pixels appartenant en même temps à la région de b et à la région de b'. Nous avons utilisé deux segmentations de nos 6 images tests, afin que leur taille diffèrent d'environ 15% en nombre de brins/sommets. Nous construirons ensuite l'appariement sur ces deux segmentations, et nous afficherons les pixels effectivement bien appariés. Dans ce jeu de données nous avons choisi de comparer des segmentations fines. L'utilisation de segmentations fines induit que nous avons plus de régions, et donc qu'il est plus difficile d'apparier correctement des brins ou des sommets. Ce qui explique pourquoi les résultats sont un peu en dessous de la moyenne des figures 6.10,6.11,6.12 et 6.13.

Nous comparons tout d'abord les résultats obtenus sans étiquettes dans la Fig. 6.16. Nous remarquons que sans les étiquettes, peu de brins et peu de sommets sont effectivement bien appariés. Si nous considérons l'étiquette basée sur l'aire des régions dans la Fig. 6.17 nous observons une nette amélioration des résultats. En effet, pour les 2G-cartes, 3 des 6 couples d'images ont plus de 25% des brins bien appariés. Pour les graphes, bien que nous observons une amélioration, elle n'est pas aussi significative. Dans la Fig. 6.18 nous ne considérons que l'étiquette indiquant la couleur moyenne des régions. Nous remarquons une seconde fois que le critère de couleur est très discriminant. Nous notons une nette amélioration avec les deux structures : pour les 2G-cartes tous les couples retrouvent plus de 25% des brins, et pour les graphes plus de 15% des sommets. Nous notons que visuellement les pixels retrouvés ont l'air aussi nombreux pour les 2G-cartes que pour les graphes, alors que la proportion de sommets et brins retrouvés est moins importante pour les graphes. Nous expliquons cette différence tout simplement parce que la méthode utilisant les graphes apparie correctement des sommets représentant des régions de plus grande taille, ainsi pour un couple de sommets retrouvés le nombre de pixels affichés est plus important. Enfin, dans la Fig. 6.19 nous utilisons les deux étiquettes. Les proportions de sommets et de brins bien appariés augmentent encore. Notez que quelle que soit la paire d'images considérée, elle est sensiblement meilleure avec la méthode utilisant les 2G-cartes.

6.2 Base de maillages

Dans cette section nous souhaitons évaluer l'intéret des 2G-cartes par rapport aux graphes pour comparer des maillages. Un maillage est un ensemble de sommets reliés par des arêtes pour former des faces. Un point de l'espace 3D est attribué à tous les sommets du maillage pour modéliser l'objet.

Modélisation de maillages

La modélisation d'un maillage avec un graphe ou une 2G-carte est directe. Le maillage est composé de sommets et d'arêtes, qui correspondront aux sommets et arêtes des graphes et des 2G-cartes. Notons que nous ne pouvons pas utiliser ici le graphe dual, associant un sommet à chaque face du maillage, et une arête à chaque relation d'adjacence entre faces, car certains maillages ont des faces ouvertes.

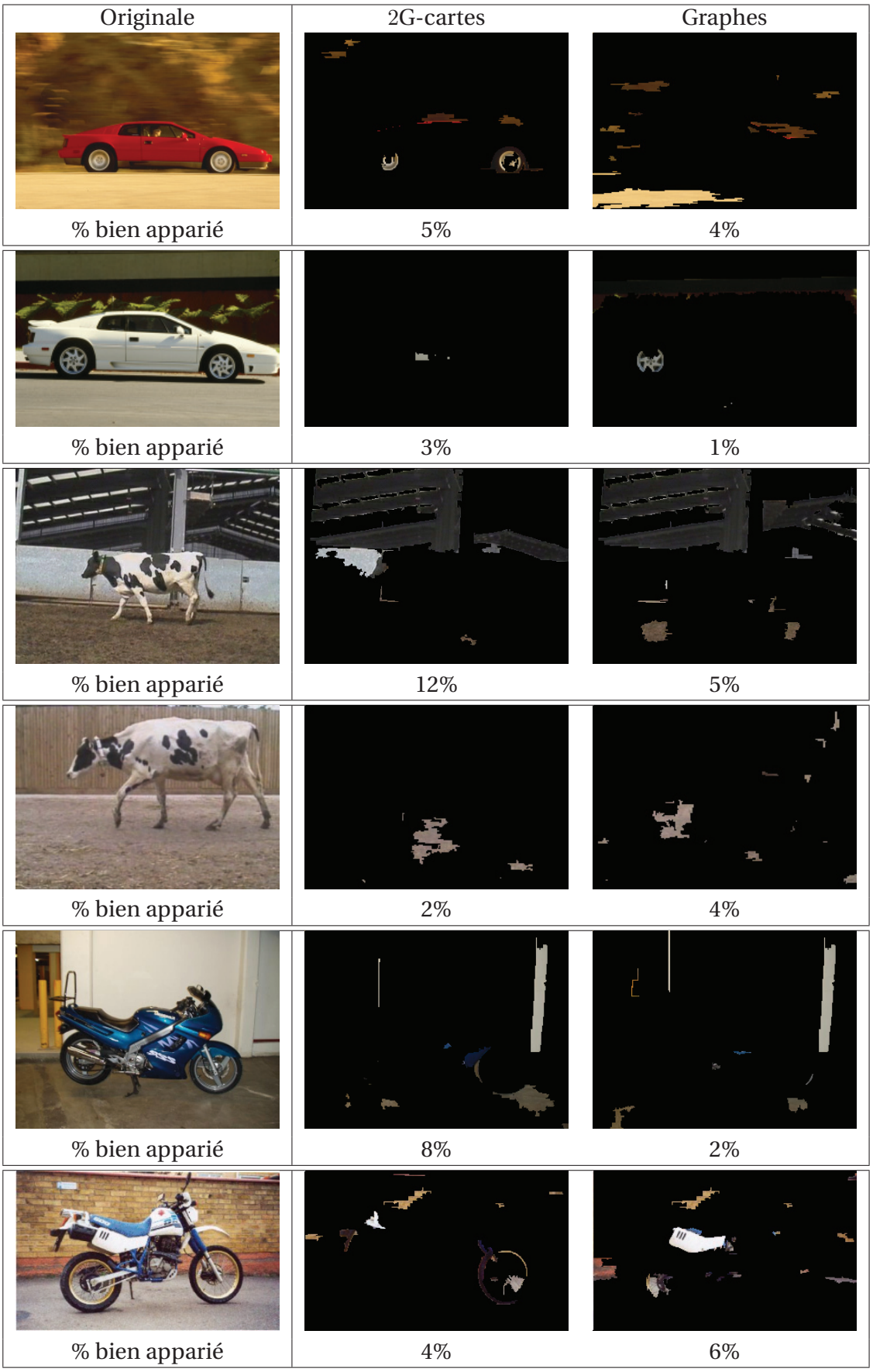


FIGURE 6.16 – *Illustration de la qualité des appariements. Pour chaque ligne, l'image de gauche est l'image originale. Les images de droite sont composées des régions pour lesquelles au moins un brin/sommet est bien apparié. Dans cette expérimentation nous n'utilisons ni l'étiquette basée sur la couleur moyenne, ni l'étiquette basée sur l'aire des régions.*

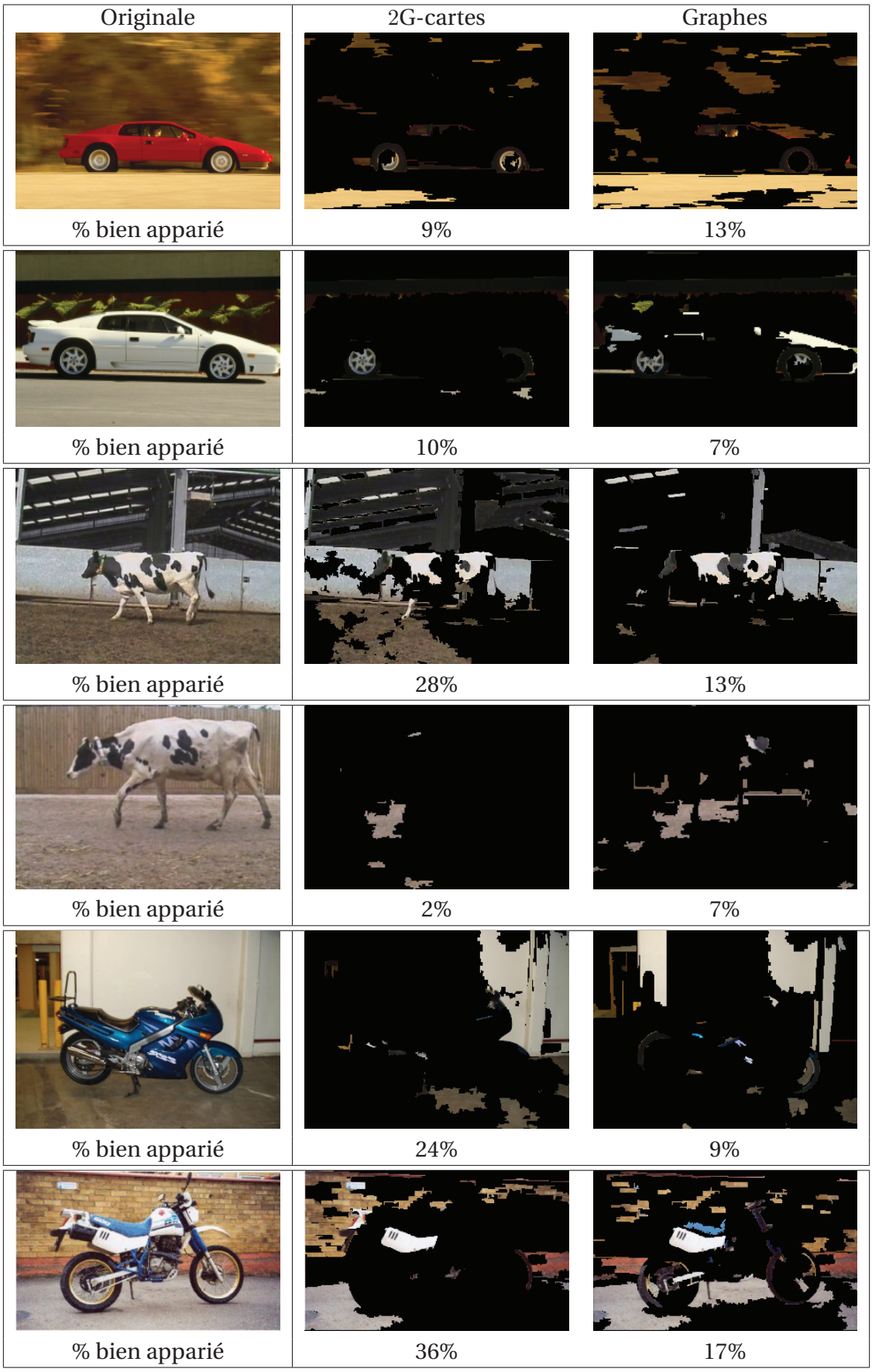


FIGURE 6.17 – *Illustration de la qualité des appariements. Pour chaque ligne, l'image de gauche est l'image originale. Les images de droite sont composées des régions pour lesquelles au moins un brin/sommet est bien apparié. Dans cette expérimentation nous utilisons les étiquettes indiquant l'aire des régions.*

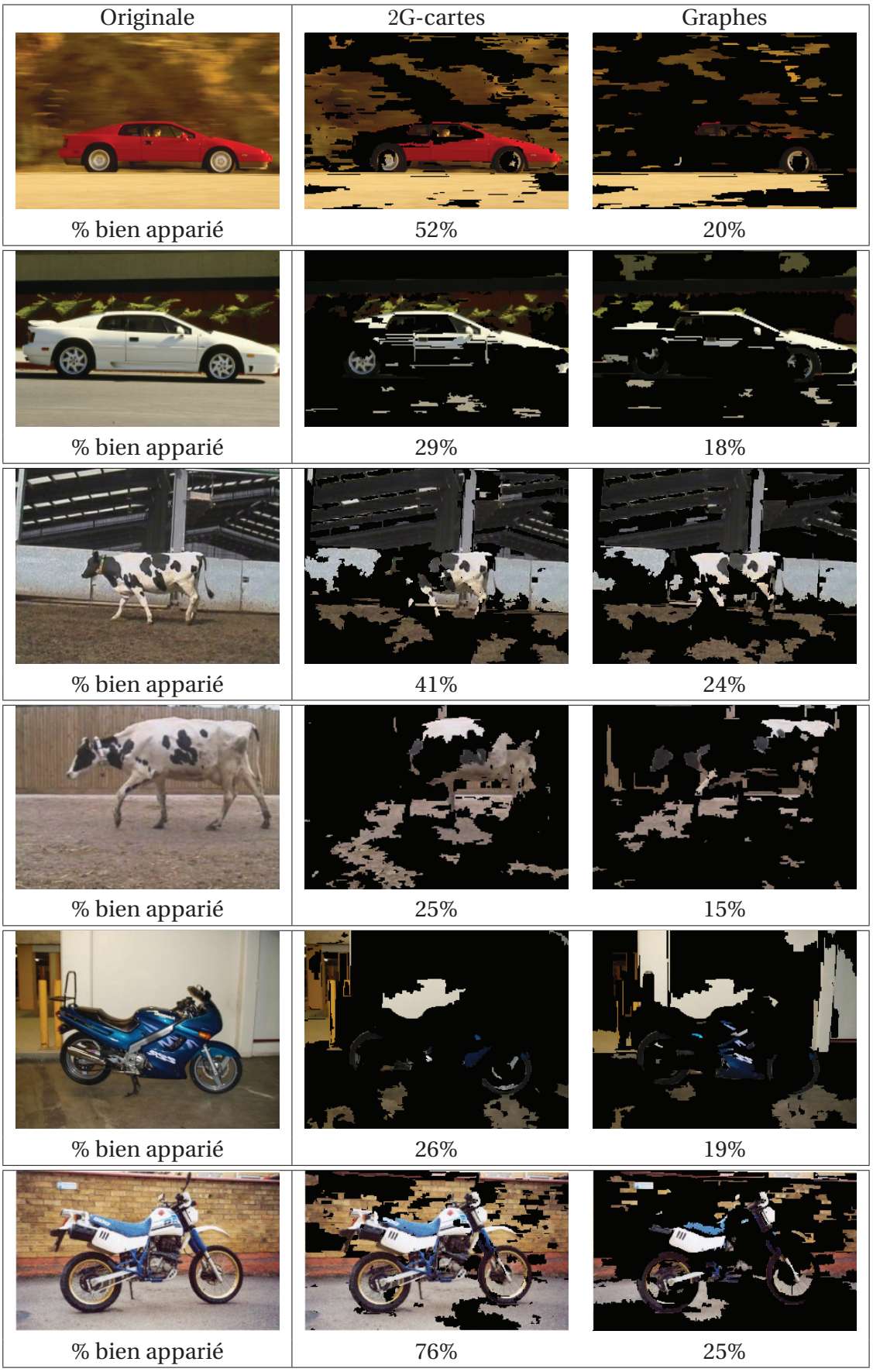


FIGURE 6.18 – *Illustration de la qualité des appariements. Pour chaque ligne, l'image de gauche est l'image originale. Les images de droite sont composées des régions pour lesquelles au moins un brin/sommet est bien apparié. Dans cette expérimentation nous utilisons l'étiquette indiquant la couleur moyenne des régions.*

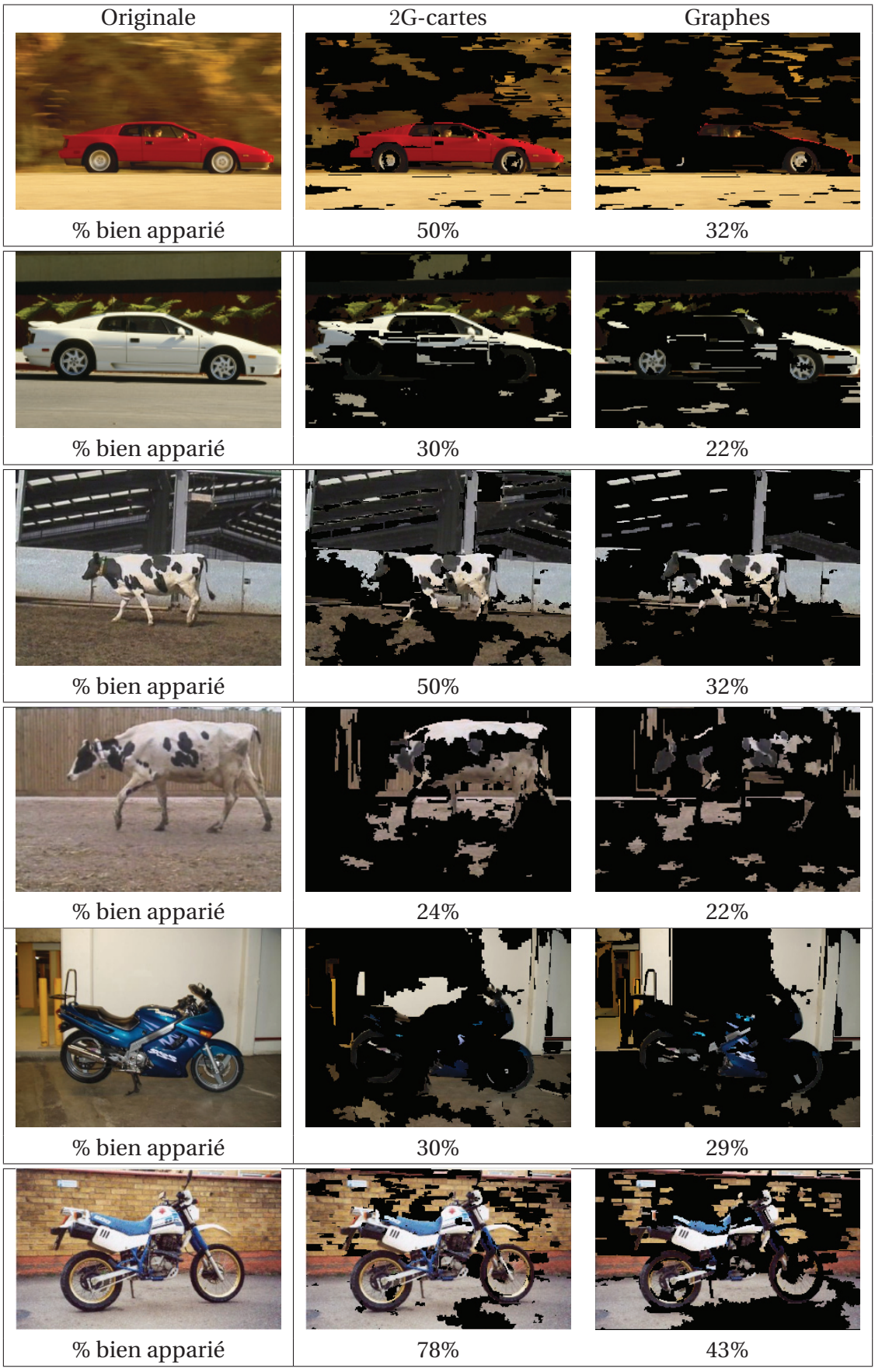


FIGURE 6.19 – *Illustration de la qualité des appariements. Pour chaque ligne, l'image de gauche est l'image originale. Les images de droite sont composées des régions pour lesquelles au moins un brin/sommet est bien apparié. Dans cette expérimentation nous utilisons la combinaison des étiquettes indiquant l'aire et la couleur moyenne des régions.*

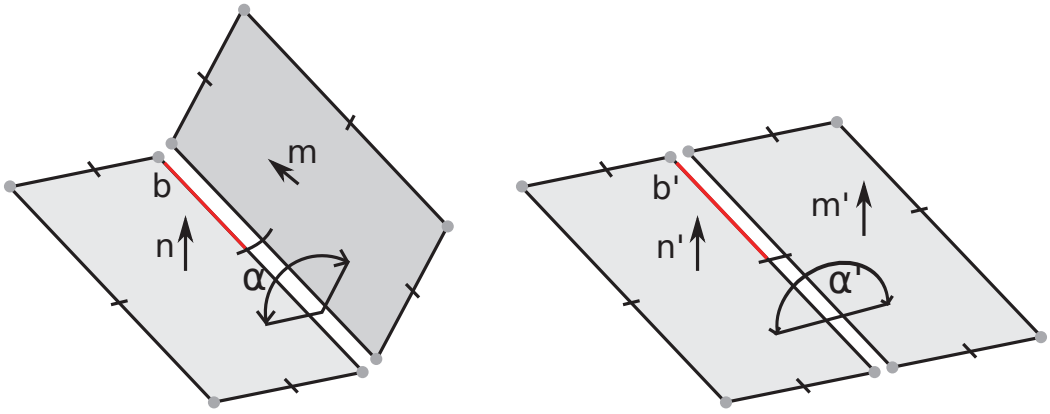


FIGURE 6.20 – *Illustration des normales des faces. Pour le brin b la normale est n et pour le brin b' la normale est n'. L'angle diédral α entre les deux faces de normales n et m, égal à l'angle entre les normales n et m, sera comparé à l'angle diédral α' égal à l'angle entre les normales n' et m'.*

6.2.1 Définition des étiquettes et coûts

Pour les maillages, les étiquettes utilisées seront différentes de celles proposées pour les images. Un maillage est composé de sommets, reliés entre eux par des arêtes, formant des faces, plongé dans un espace en 3 dimensions. Chaque face possède alors une orientation, qui est représentée par son vecteur normal. Notre première heuristique sera basée sur l'orientation des faces. Cependant notre méthode pour modéliser un maillage avec un graphe ne permettant pas de modéliser les faces, nous ne pourrons pas considérer cette heuristique. Nous proposons ici une heuristique basée sur l'angle de l'arête entre ses deux faces adjacentes. Cette information peut être portée par une étiquette sur l'arête, cette heuristique sera donc utilisable autant pour les graphes que pour les 2G-cartes. Enfin, nous ne travaillons pas uniquement avec des maillages triangulaires. Les faces ne sont donc pas forcément triangulaires et donc pas toujours planaires. L'aire des faces n'étant donc pas calculable, nous proposons une dernière heuristique basée sur le périmètre des faces.

Orientation des faces

L'utilisation des normales (vecteurs orthogonaux aux faces) est un bon critère géométrique qui va nous permettre de caractériser l'orientation des faces et ainsi favoriser l'appariement de brins faisant partie de faces ayant la même orientation (normales similaires).

Pour cela, chaque brin est étiqueté par la normale de sa face, et le calcul du coût de substitution de cette étiquette entre un brin b incident à une face de normale n et un brin b' incident à une face de normale n' sera :

$$c_{subst-normale}(b, b') = acos(n \cdot n') \tag{6.3}$$

Ainsi, le coût de substitution est égal à 0 dans le cas où les faces possèdent la même normale, et à 1 si les normales sont opposées. Notez que pour des cartes représentant des maillages non triangulaires, les faces ne sont pas toujours planaires. Nous résolvons ce problème en utilisant des normales moyennes aux faces.

La Fig. 6.20 illustre l'appariement de deux brins b et b' (en rouge) ayant pour normales respectives n et n'. Le coût de substitution sera d'autant plus faible que l'angle

entre les deux normales sera faible. Il sera minimum dans le cas où les normales seront égales et le coût sera de 1 dans le cas de normales opposées.

Angle diédral

La normale permettant de décrire l'orientation n'est pas l'unique information géométrique que contient un maillage. Nous proposons donc une étiquette pour décrire la courbure d'une zone en utilisant l'angle diédral.

L'angle diédral est l'angle entre deux faces adjacentes. Cette information va permettre de favoriser l'appariement des zones planes avec d'autres zones planes, et des zones à forte courbure avec d'autres zones à forte courbure. Sur chaque brin, une étiquette est ajoutée portant l'angle diédral de la face qui lui est incidente avec la face qui lui est adjacente en ce brin.

Dans la Fig. 6.20 l'angle α est l'angle diédral entre les deux faces de normales n et m. Il est associé au brin b. L'angle α' est l'angle diédral des faces de normales n' et m'. Il est associé au brin b'.

Pour les graphes, nous associons cette étiquette aux arêtes.

Le coût de susbstitution de deux étiquettes l et l' sera égal à :

$$c_{subst-angle}(l,l') = \frac{max(\alpha,\alpha') - min(\alpha,\alpha')}{180} \tag{6.4}$$

Ce coût est égal à 0 dans le cas où les angles sont identiques, et 1 si les angles sont opposés (avec une différence de 180).

Le coût de substitution de deux brins ou de deux arêtes est égal au coût de substitution de leurs étiquettes. Enfin, il est possible, comme pour les images, de tenir compte de l'aire des faces, dans le cas où toutes les faces sont planaires et fermées. Dans le cas où les cartes possèdent des faces non planaires et/ou non fermées l'aire n'est pas calculable. Pour cela nous proposons une nouvelle étiquette basée sur le périmètre des faces.

Périmètre des faces

Sur chaque brin, une étiquette est ajoutée contenant le périmètre de la face incidente au brin. Le coût de substitution du brin b incident à une face ayant pour périmètre P par le brin b' incident à une face de périmètre P' est égal à :

$$c_{subst-perim}(b,b') = 1 - \frac{min(P,P')}{max(P,P')} \tag{6.5}$$

Le coût de substitution sera ainsi d'autant plus grand que le ratio entre le périmètre des faces est grand.

Définition des coûts d'édition

Pour ces expérimentations nous avons choisi des coûts pour l'ensemble des opérations d'éditions. Pour les graphes, nous avons ajouté un coût de substitution $c_{subst-angle}$ entre les étiquettes portant l'information de l'angle diédral. Nous avons aussi défini un coût d'ajout ou de suppression de sommet égal à 1, un coût d'ajout ou de suppression d'arêtes égal à 1 et un coût de substitution nul entre les sommets ces derniers n'ayant pas d'étiquettes structurelles.

Pour les 2G-cartes, nous utilisons les coûts proposés $c^p_{1,1}$ dans la Section 4.5. Pour rappel nous avons donc, pour deux cartes comparées G et G' :
- $c^p_{1,1}(del_E) = |E||\{(d_1, i, d_2) \in coutures_{G'}(D'), d_i \in E \text{ où } d_2 \in E\}|)$;
- $c^p_{1,1}(add_{E,F}) = |E| + |F|$;
- $c^p_{1,1}(sew_F) = c^p_{1,1}(unsew_F) = |F|$.

Pour le coût de substitution entre brins, nous considérons une somme des coûts de substitution des étiquettes $c_{subst-angle}$, $c_{subst-perim}$ et $c_{subst-normale}$.

6.2.2 Description des jeux de données

Les maillages triangulaires sont des maillages réguliers pour lesquels toutes les faces sont des triangles. Un des principaux avantages à utiliser des structures telles que les 2G-cartes est que nous pouvons nous affranchir d'une limitation sur le nombre d'arêtes par faces. Il est ainsi possible de traiter des maillages avec des quadrilatères, ou des polygones quelconques en guise de faces. Dans le cas d'un maillage triangulaire, pour représenter un polygone de k côtés il faut le trianguler et utiliser $k-2$ triangles ce qui augmente le nombre de primitives pour décrire un objet. Nous nous proposons d'étudier notre algorithme sur des maillages non triangulaires, de commencer par évaluer nos heuristiques sur une application de classification de maillages pour ensuite évaluer la qualité de l'appariement construit.

MeshClassification

Pour le jeu de données *MeshClassification*, l'objectif est de montrer que nous pouvons effectuer de la classification de maillages avec des approches structurelles. Nous avons sélectionné quatre maillages (Fig. 6.21) issus de la base [AIM]. Dans un premier temps, nous avons fusionné toutes les faces ayant un angle avec leur face adjacente de moins de 5°. Nous obtenons ainsi quatre maillages ayant de plus grandes faces dans les zones planes. Les maillages ainsi créés sont illustrés dans la Fig. 6.21. Nous avons ensuite modélisé ces maillages avec des 2G-cartes, puis dégradé ces 2G-cartes en 10 2G-cartes de plus petites tailles pour obtenir 4 classes de 10 2G-cartes. Pour dégrader les 2G-cartes nous avons utilisé l'opération de suppression de brins del et l'opération de décousure $unsew$ aléatoirement jusqu'à obtenir des 2G-cartes ayant une taille finale entre 15% et 25% de la 2G-carte originale. Après l'application de chaque opération, nous décousons les brins violant les contraintes d'intégrité afin d'obtenir une 2G-carte valide. Nous obtenons donc 4 classes correspondant aux quatre maillages composées des 10 2G-cartes dégradées toutes ayant un nombre de brins similaires. Nous avons aussi modélisé ces 2G-cartes dégradées avec des graphes. Nous modélisons simplement chaque sommet de la 2G-carte par un sommet du graphe, et nous ajoutons une arête entre deux sommets du graphe s'il existe une arête dans la 2G-carte. Les opérations del et $unsew$ mènent souvent à des faces ouvertes, nous ne pouvons donc pas utiliser une représentation avec un graphe dual.

Quelques statistiques sont présentées dans le Tableau 6.5. Nous remarquons que les 2G-cartes générées ont toutes un nombre de brins et de coutures très similaires, par contre le nombre de sommets, d'arêtes et de faces sont très différents. Notez que les nombres de composantes connexes diffèrent également. En effet, certains maillages sont composés de différentes composantes connexes, comme pour le maillage modélisant le téléphone pour lequel les touches sont toutes dissociées. Pour les autres maillages ce sont principalement les opérations $unsew$ et del utilisées qui créent les

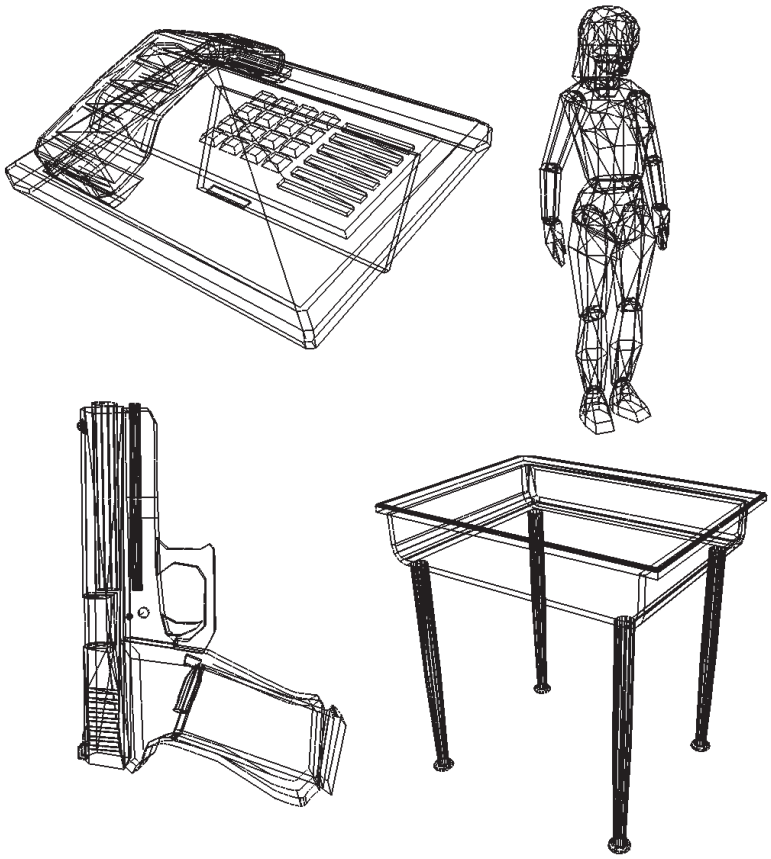


FIGURE 6.21 – *Maillages originaux utilisés dans le processus de classification.*

Classe	#brins
	moy(min max)
		#coutures
		moy(min max)
			#sommets
			moy(min max)
				#arêtes
				moy(min max)
					#faces
					moy(min max)

Human	4466 (4354 4556)	10935 (10066 11784)	965 (817 1129)	1687 (1573 1786)	888 (806 981)
Phone	4479 (4358 4563)	11013 (10060 11784)	1159 (1041 1283)	1680 (1571 1793)	722 (593 859)
Pistolet	4467 (4356 4533)	10219 (9242 10706)	1432 (1359 1552)	1877 (1821 1983)	799 (708 995)
Table	4466 (4396 4557)	10635 (10050 11386)	1096 (968 1194)	1767 (1676 1844)	882 (801 964)

Tableau 6.5 – *Étude du jeu de données utilisé pour le processus de classification.*

Classe	#brins moy(min max)	#coutures moy(min max)	#sommets	#arêtes moy(min max)	#faces moy(min max)
Arm	5116 (3628 7260)	20415 (14460 28988)	323	646 (460 914)	324 (138 592)
ArmChair	5511 (4228 8268)	16518 (12668 24788)	830	1381 (1061 2071)	552 (232 1242)
Bird	5645 (4188 8352)	16937 (12564 25056)	754	1411 (1047 2088)	659 (295 1336)
Body	4878 (3176 7660)	14502 (9396 22848)	700	1252 (827 1948)	549 (124 1245)
Buste	4784 (3138 6834)	18803 (12234 26994)	312	639 (432 897)	326 (120 583)
Head	7408 (5432 9984)	29456 (21552 39760)	452	948 (701 1270)	494 (247 816)
Lamp	7019 (5212 8524)	21057 (15636 25572)	1128	1754 (1303 2131)	634 (183 1011)
Plane	5228 (4036 7788)	15412 (11836 23092)	976	1375 (1077 2015)	399 (101 1039)

Tableau 6.6 – *Étude du jeu de données utilisé pour l'étude de la qualité de l'appariement.*

différentes composantes connexes.

MeshQuality

L'objectif de cet expérimentation est d'étudier la qualité des appariements construits pour comparer des graphes et 2G-cartes. Pour cela, nous avons sélectionné 8 maillages de la base [AIM] (illustré dans la Fig. 6.22) que nous avons dégradés. La dégradation consiste en la fusion de faces dont l'angle diédral est inférieur à une valeur θ. Nous obtenons ainsi des maillages ayant des faces avec plus d'arêtes dans les zones planes. Pour générer nos classes nous avons fait varier θ de 5 à 30 et nous avons conservé tous les maillages modélisés par des 2G-cartes ayant un nombre de brins compris entre 3000 et 10000. Quelques statistiques sur ce jeu de données sont présentées dans le Tableau 6.6.

Notez que les maillages dégradés ont tous le même nombre de sommets que le maillage original. En effet, l'opération de fusion de faces ne supprime pas de sommets.

6.2.3 Classification de maillages

Protocole expérimental

Nous avons classé les maillages du jeu d'essai $MeshClassification$ en utilisant l'algorithme des k plus proche voisins (k-nn) selon un principe 'leave-one-out'. Comme pour les images, nous avons exécuté notre algorithme avec différentes combinaisons d'étiquettes. Nous avons commencé par ne tenir compte d'aucune étiquette pour évaluer la capacité à classifier un tel jeu de données uniquement en utilisant la structure, puis nous avons intégré dans un second temps les trois étiquettes proposées.

Résultats

La Fig. 6.23 montre les résultats de classification du jeu de données sans aucune étiquettes. Nous notons l'algorithme calculant une approximation de la distance d'édition de graphes obtient de bien bons résultats que notre algorithme sur les cartes généralisées. Nous avons ensuite exécuté sur le même jeu de données l'algorithme de classification en ajoutant les étiquettes. Les deux algorithmes, sans distinction et indépendemment de l'étiquette considérée, classifient 100% du jeu de données. Les étiquettes sont donc particulièrement discriminantes sur ce jeu d'essai.

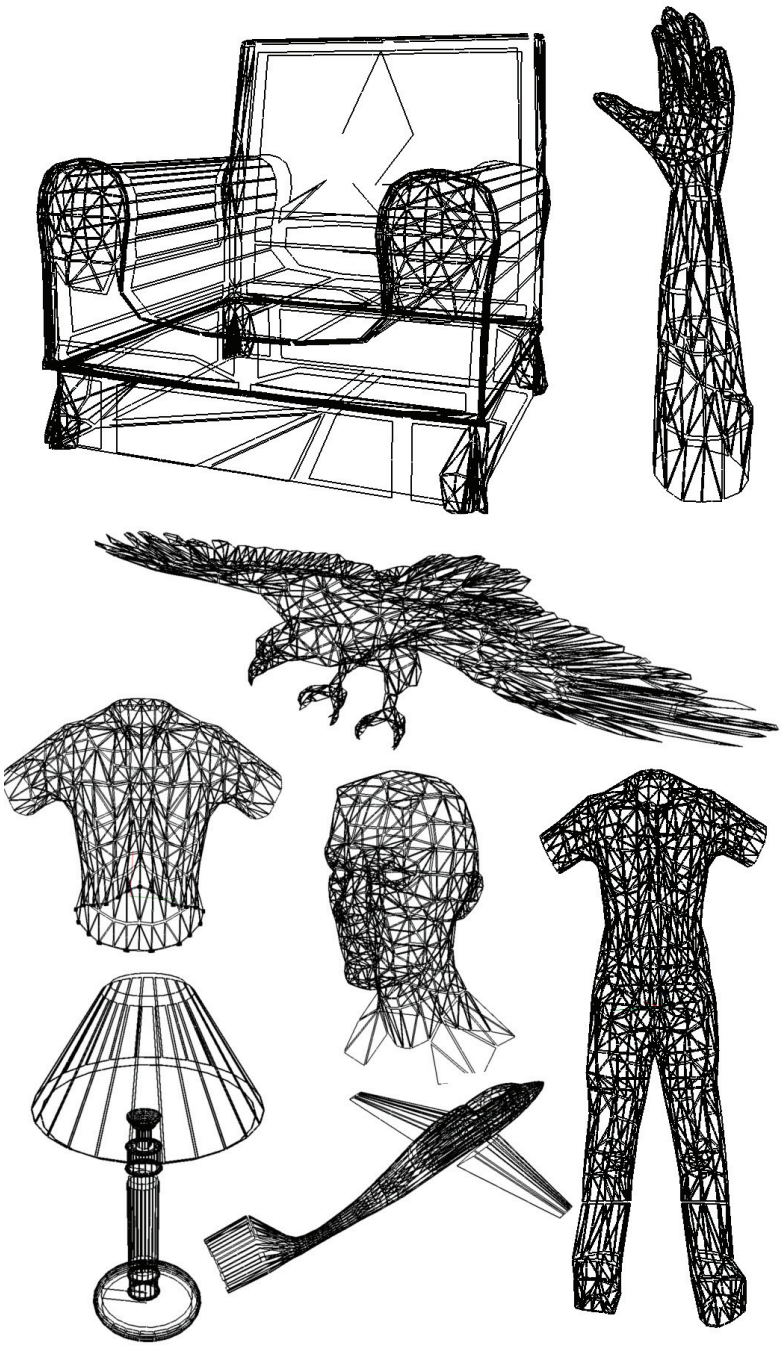


FIGURE 6.22 – *Exemples de maillages dégradés utilisés pour le processus de mesure de qualité de l'appariement.*

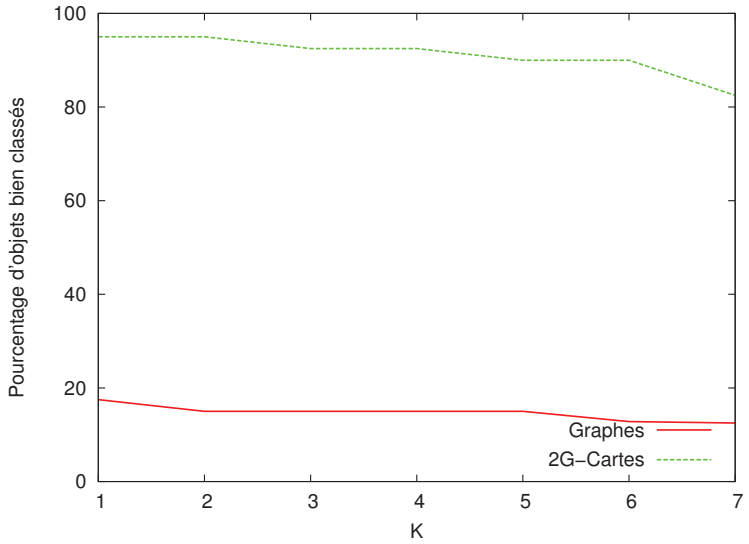


FIGURE 6.23 – *Taux de bonne classification en fonction de la valeur k de l'algorithme de classification k-nn. Les courbes représentent les taux de bonne classification pour notre algorithme (2G-carte) et l'algorithme [RB09] (Graphes) sans étiquettes.*

6.2.4 Comparaison des appariements construits

Comme nous l'avons déjà montré sur les images, notre algorithme ne calcule pas seulement une distance mais aussi un appariement entre les brins des nG-cartes comparées, de même pour l'algorithme de [RB09] qui apparie les sommets des graphes comparés. Dans cette expérimentation nous allons comparer la proportion de brins et de sommets bien appariés par rapport à la dégradation appliquée sur un maillage.

Protocole expérimental

Dans cette étude nous utilisons le jeu de données $MeshQuality$ en comparant indépendemment toutes les classes de maillages entre elles. Puis nous avons exécuté les deux algorithmes calculant les appariements entre graphes et entre 2G-cartes. Nous avons alors étudié chaque appariement ainsi créé et relevé la proportion de sommets et de brins bien appariés. Deux sommets de graphes sont dits bien appariés s'ils apparient deux sommets du maillage ayant les mêmes coordonnées 3D. Dans le cas des brins, deux brins sont dits bien appariés si leurs deux sommets incidents ont les mêmes coordonnées 3D. Enfin, comme le nombre de sommets n'est pas modifié par l'opération de fusion de faces, nous représentons la proportion des brins et sommets bien appariés par rapport à la dégradation des 2G-cartes en nombre de brins. Pour notre algorithme nous avons utilisé les trois heuristiques h_{cout}, h_{freq} et h^{+}_{deg}, et retenus à chaque fois le meilleur appariement sur 10 exécutions de $ApproxD$.

Résultats

Les résultats sont illustrés dans 2 graphiques, le premier effectuant la comparaison entre les 2G-cartes et les graphes, et le second donnant tous les résultats obtenus pour les 2G-cartes. Dans la Fig. 6.24, nous montrons que la proportion de sommets bien appariés avec les graphes, sans aucune étiquette est trop faible. En ajoutant l'étiquette

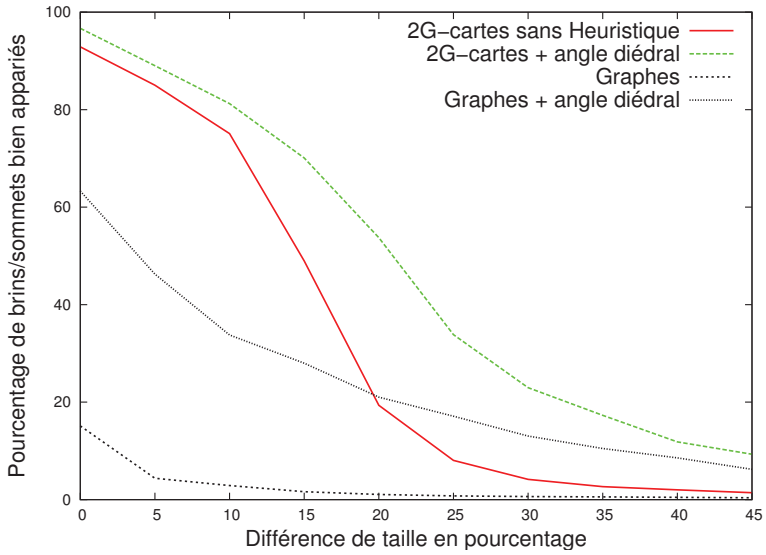


FIGURE 6.24 – *Proportion de brins et sommets bien appariés en fonction du taux de dégradation (en pourcentage de brins) de la 2G-carte modélisant le maillage. Les courbes représentent les résultats obtenus en fonction des différentes étiquettes considérées avec les graphes et les 2G-cartes.*

de l'angle diédral sur les arêtes la proportion de brins bien appariés augmente. Cependant les résultats sont moins bons que ceux obtenus par notre algorithme sur les 2G-cartes. Cela est probablement du à un manque d'information dans notre modélisation de maillages à l'aide de graphes par rapport à la modélisation de maillages à l'aide de 2G-cartes. Dans les deux cas, la proportion de brins bien appariés est en dessous de 20% dès que la taille des 2G-cartes modélisant les maillages diffère de 40%. Notez que quand la dégradation est de 0% c'est-à-dire que nous comparons des 2G-cartes et des graphes isomorphes, la méthode utilisant les graphes permet d'apparier seulement 18% des sommets. L'appariement calculé par l'algorithme n'associe donc pas les sommets avec les bons sommets correspondants. Cela est du au fait que l'algorithme ne considère les choix que localement.

Dans la Fig. 6.25 nous comparons la proportion de brins bien appariés en fonction de la dégradation et des étiquettes considérées. Notez que l'étiquette indiquant l'orientation et l'angle diédral permet l'obtention de meilleurs résultats. La combinaison des trois étiquettes améliore encore d'avantage la proportion de brins bien appariés.

Nous avons représenté dans la Fig. 6.26 des exemples de couples de maillages comparés avec une partie de l'appariement calculé avec notre algorithme.

Notez que nous n'utilisons pas la position des sommets dans nos heuristiques, ceci nous permet de réaliser l'appariement quelle que soit la position du maillage dans son système de coordonnées. Ensuite, il est intéressant de noter que les étiquettes proposées ne sont pas toutes résistantes aux transformations rigides. Pour le montrer, nous avons exécuté notre algorithme sur deux ensembles de 8 paires de maillages. Pour les 8 premières paires, nous comparons deux 2G-cartes isomophes qui diffèrent d'une transformation rigide. L'ensemble des transformations considérées sont :

– rotation de 180° selon l'axe (1,1,1)
– homothétie de coefficient 2
– translation par le vecteur (1,1,1)

Le Tableau 6.7 présente les résultats obtenus lors de cette comparaison. Nous n'avons ici pas tenu compte de la substitution des étiquettes dans le calcul des dis-

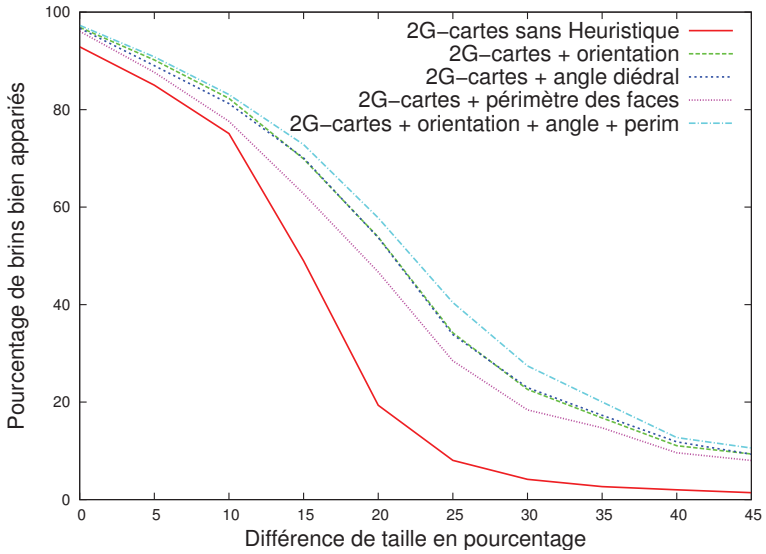


FIGURE 6.25 – *Proportion de brins biens appariés en fonction du taux de dégradation (en pourcentage de brins) de la 2G-carte modélisant le maillage. Les courbes représentent les résultats obtenus en fonction des différentes étiquettes considérées pour les 2G-cartes.*

Déformation	Sans Heuristiques	Orientation	Angle diédral	Perimètre
Sans Tranformations	0	0	0	0
Rotation	0	1151	0	0
Homothétie	0	0	0	7895
Translation	4	0	0	6

Tableau 6.7 – *Moyenne des distances d'édition calculées entre les deux 2G-cartes isomorphes de 8 paires de maillages. L'une des deux 2G-cartes a subi une transformation géométrique rigide.*

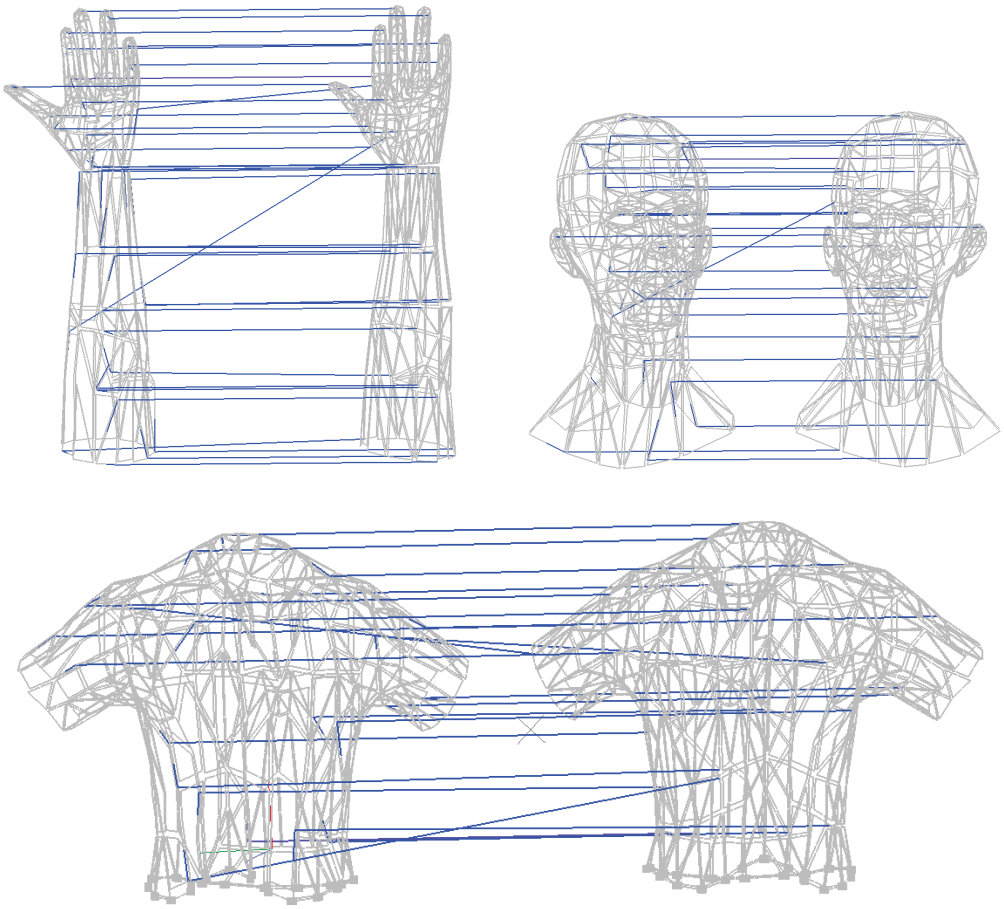


FIGURE 6.26 – *Exemples d'appariements obtenus avec notre algorithme sur des 2G-cartes sans utilisation de la géométrie. Les 2G-cartes modélisent des maillages triangulaires pour lesquels les faces coplanaires ont été fusionnées. La variation de l'angle de tolérance de cette fusion permet l'obtention de maillages de différentes tailles. Les appariements de brins qui sont visualisés en bleu ont été choisis aléatoirement.*

Déformation	Sans Heuristiques	Orientation	Angle diédral	Perimètre
Sans Tranformations	12329	11807	10959	12253
Rotation	12309	12459	11150	12220
Homothétie	12103	11733	10860	13657
Translation	11976	11863	10887	12132

Tableau 6.8 – *Moyenne des distances d'édition calculées entre les deux 2G-cartes qui diffèrent en moyenne de 20% en nombre de brins de 8 paires de maillages. L'une des deux 2G-cartes a subi une transformation géométrique rigide.*

tances. En conséquence, dans le meilleur des cas, les 2G-cartes étant isomorphes, nous devrions obtenir des distances nulles. Cependant, notez que l'étiquette 'Orientation' dans le cas de la rotation ne permet pas de retrouver l'appariement décrivant l'isomorphisme. En effet, en moyenne le chemin d'édition calculé nécessite 1151 modifications de brins ou coutures pour modifier une des deux 2G-cartes en la seconde. Le résultat est plutôt cohérent, car cette heuristique va diriger la construction de l'appariement en associant les faces tournées du même côté, ce qui n'est pas la solution optimale étant donné que le maillage a subi une rotation. De la même manière, l'étiquette indiquant le périmètre des faces n'est pas robuste par rapport au changement d'échelle.

Pour la seconde série de 8 paires, nous avons utilisé 8 couples de 2G-cartes de l'expérimentation précédente ayant une taille qui diffère de 20% en moyenne. Nous avons ensuite exécuté notre algorithme avec nos trois heuristiques séparement, les résultats sont illustrés dans le Tableau 6.8. Notez que comme pour le cas isomorphe, les étiquettes indiquant l'orientation des faces et le périmètre des faces ne sont respectivement pas robustes face aux rotations et homothéties. Dans cette expérimentation l'objectif est d'obtenir la plus petite valeur en moyenne de l'approximation de la distance d'édition. Nous notons que l'angle diédral se distingue cette fois du calcul de la distance sans considérer d'étiquettes en ayant de meilleurs résultats. Cette étiquette basée sur l'angle entre deux faces est donc une étiquette robuste pour le calcul de notre distance sur des 2G-cartes modélisant des maillages.

6.3 Discussion

Nous avons présenté des expérimentations préliminaires de notre algorithme pour calculer une approximation supérieure de notre distance. Nous avons étudié le comportement de notre algorithme sur des 2G-cartes modélisant des images. Nous nous sommes comparés à un des algorithmes les plus connus pour le calcul de la distance d'édition de graphes. Nous avons montré que leur algorithme est meilleur pour classifier les images de notre jeu de tests dès lors qu'il exploite les informations colorimétrique. Cependant, il reste moins bon qu'un algorithme de classification de type histogramme basé sur la colorimétrie. Nous notons aussi que sans les étiquettes, l'algorithme basé sur les graphes n'arrive pas du tout à classifier notre jeu de données. Dans une seconde expérimentation nous avons comparé la qualité de l'appariement construit par notre algorithme à celui obtenu dans le cas de graphes. Nous avons montré que le nombre d'appariements cohérents est plus important en proportion pour notre algorithme que pour les graphes. Nous avons aussi illustré ce résultat sur six images en montrant l'utilité des étiquettes colorimétriques et des étiquettes basées sur l'aire des régions. Enfin nous nous sommes attelés à l'utilisation de notre algorithme pour comparer des maillages. Nous avons tout d'abord effectué une classification de

maillages non triangulaires dégradés et ensuite étudié la proportion d'appariements cohérents. Dans l'ensemble, les résultats sont plutôt encourageants mais ne sont que des résultats préliminaires. Nous avons longtemps cherché une application directe à la mesure de similarité de 2G-cartes mais nous n'avons pas trouvé d'applications pour lesquelles nous sommes réellement compétitifs quand nous nous comparons à l'état de l'art spécialisé. Cela est principalement du à la modélisation des objets comparés. Il manque encore un travail en amont sur la meilleure méthode pour représenter certains objets à base de nG-cartes. Aussi notez que notre algorithme est un algorithme de type glouton qui est extrêmement dépendant de son point de départ. L'algorithme a été principalement développé pour valider la méthode et les définitions théoriques afin de montrer qu'il est possible d'observer les mesures de similarité différemment avec les nG-cartes que nous le faisons actuellement avec les graphes. Il serait intéressant de développer un algorithme plus global pour être moins dépendant de ce point de départ.

Chapitre 7

Conclusion

Bilan

Nous avons proposé dans cette thèse des méthodes permettant de mesurer la similarité entre cartes généralisées. Dans un premier temps nous avons défini un ensemble de concepts ainsi qu'une distance basée sur la plus grande sous-carte commune. Dans un second temps nous avons défini un ensemble d'opérations pour modifier des cartes généralisées et une distance d'édition basée sur le nombre d'opérations minimal pour modifier une carte généralisée en une autre. Ensuite nous avons proposé un algorithme glouton et un ensemble d'heuristiques construisant un appariement de brins. Nous avons montré que cet appariement permet d'extraire une sous-carte commune, ce qui nous donne un moyen pour calculer une borne de nos distances. Nous avons validé l'algorithme et nos heuristiques sur un jeu de données généré. Enfin, nous avons proposé un ensemble d'étiquettes spécialisées dans la description d'images et de maillages et nous avons étudié leur impact sur un test de classification et de mise en correspondance des éléments d'une base d'images et d'un ensemble de maillages.

Distances Nous avons tout d'abord défini deux types de sous-cartes, les sous-cartes induites et les sous-cartes partielles. Pour plus de flexibilité les sous-cartes partielles autorisent que les coutures ne soient pas toutes conservées. Elles permettent donc d'obtenir des sous-cartes de plus grande taille, notamment en permettant d'apparier plus de brins. Nous avons défini la taille d'une nG-carte avec une pondération ω_1 et ω_2 sur le nombre de brins et le nombre de coutures afin de pouvoir modifier l'influence des coutures par rapport aux brins. En pratique, ces deux pondérations sont intimement liées : c'est en retrouvant le plus de coutures que nous retrouvons le plus de brins. En effet, dans le cas induit, en commençant par choisir les couples de brins qui retrouvent le plus de coutures, nous réduisons le nombre de libertés des brins appariés. Nous avons donc moins de chance de choisir ensuite des couples de brins ne retrouvant pas de coutures. Nous avons ensuite proposé un ensemble d'opérations modifiant les cartes généralisées. Nous avions tout d'abord pensé utiliser les opérations d'insertion et de suppression de cellules définies dans [SDL06], mais nous n'avons pas su définir un appariement de brins ou de cellules décrivant une unique suite d'opérations pour transformer une nG-carte en une autre. Nous avons donc décidé de proposer des opérations basiques modifiant directement les brins et les coutures de nos cartes généralisées. Ces opérations doivent par contre être manipulées avec attention, car il est possible de définir des opérations modifiant une carte généralisée valide en un ensemble de brins et de coutures n'étant plus une carte généralisée valide. Enfin, notez que nous avons

montré que le chemin d'édition calculé à partir de la plus grande sous-carte commune permet de calculer une distance minimale si nous considérons des suites d'opérations valides. Nous n'avons pas su montrer ni trouver de contre-exemple de l'existence d'un chemin d'édition potentiellement plus court en considérant des opérations passant par des nG-cartes intermédiaires invalides.

Algorithme et heuristiques Nous avons proposé un algorithme permettant de calculer une approximation de la plus grande sous-carte commune. Étant donné la NP-complétude du problème nous avons opté pour un algorithme glouton simple permettant d'obtenir un premier résultat afin de valider la théorie. Nous avons ensuite proposé des améliorations de cet algorithme en proposant des heuristiques génériques basées sur des informations de voisinage de la nG-carte ainsi que sur des étiquettes adaptées aux images et maillages. Nous avons montré dans la Section 5.3 que l'algorithme est extrêmement dépendant de son point de départ. En effet, une fois un couple de départ choisi l'heuristique h_{cout} oriente le choix des appariements pour retrouver un maximum de coutures en propageant l'appariement autour de son point de départ. Le choix du point de départ est donc primordial. Nous avons alors proposé un ensemble d'heuristiques s'appuyant sur la description du voisinage des brins. h_{deg} et h^{+}_{deg} décrivent le voisinage des cellules incidentes et adjacentes aux brins. Nous avons montré dans le chapitre d'expérimentation que les heuristiques utilisant les degrés et co-degrés des brins permettent d'obtenir une meilleure approximation de notre plus grande sous-carte commune. Le principal inconvénient de cet algorithme est qu'il calcule un appariement localement. Or nous pouvons créer des nG-cartes pour lesquelles la plus grande sous-carte ne pourrait être obtenue avec notre algorithme qu'en considérant plusieurs graines simultanément. En effet, dans l'état actuel, la propagation de l'appariement s'effectue jusqu'à ce qu'il n'existe plus de couples de brins retrouvant de nouvelles coutures. Il aura donc très peu de chances de trouver la meilleure plus grande sous-carte commune.

Expérimentations Nous avons validé notre algorithme sur deux bases : la première base est une base d'images et la seconde une base de maillages. Dans les deux cas nous nous sommes comparés à la distance d'édition entre graphes. Pour cela, nous avons proposé des méthodes pour modéliser des images et des maillages au moyen de graphes et de 2G-cartes. Or la capacité de représentation d'un graphe est différente de la capacité de représentation des 2G-cartes, nous ne comparons donc pas des objets équivalents. Nous notons aussi une différence de grandeur, les cartes ayant en moyenne dix fois plus de brins que les graphes n'ont de sommets. Pendant ces expérimentations préliminaires, nous avons noté que les graphes sans les étiquettes ne donnent pas de bons résultats. Mais, en contrepartie, ils exploitent mieux les étiquettes quand elles sont disponibles. Ainsi, pour classifier des images, la colorimétrie est particulièrement importante et donc la méthode utilisant les graphes obtient de meilleurs résultats. Cependant, si nous observons la proportion des régions bien appariées par ce dernier, elle est plus faible que la proportion des régions bien appariés par notre algorithme. Concernant les maillages, nous notons aussi que les zones appariées par notre algorithme sont relativement cohérentes, ce qui n'est pas le cas des graphes. C'est sans doute dû au fait que nous considérons un graphe primal ayant trop peu d'informations mais ce serait à vérifier. Nous avons aussi montré que construire un appariement en se basant sur la structure permet d'être relativement tolérant aux transformations rigides, mais d'un autre côté, il n'est pas résistant aux forts changements de structures. Par exemple la comparaison de deux objets identiques mais avec un échantillonage différent n'obtiendra pas de bons résultats. En effet, dans ce cas il est beaucoup plus simple de comparer

la géométrie de ces objets plutôt que leur structure. Notez aussi que nous avons travaillé sur des maillages non triangulaires. Il est tout à fait possible de travailler sur des maillages triangulaires mais il faut alors considérer des étiquettes géométriques pour palier le comportement de l'heuristique h^+_{deg} dans le cas de régularités.

Discussion sur les méthodes structurelles Les méthodes structurelles telles que les chaînes, les arbres, les graphes et les cartes sont des structures permettant de modéliser des objets. Pour mesurer la similarité entre deux objets nous pouvons alors mesurer la similarité de ces structures. Dans certains cas particuliers, comme par exemple les chaînes pour les brins d'ADN et les graphes pour les molécules, elles sont parfaitement adaptées. Mais dans tous les cas, il semble intéressant de comparer également des étiquettes associées aux structures. Dans d'autres, il existe une infinité de méthodes pour modéliser des objets, il convient donc de trouver une méthode de modélisation telle que la distance entre les structures soit représentative de la distance entre les objets. L'objectif de cette thèse étant de montrer que les cartes apportent de l'information supplémentaire par rapport aux graphes, et que cette information peut être exploitée pour mesurer la similarité. Nous avons montré que les cartes, à l'instar des graphes, peuvent être utilisés pour classifier et mettre en correspondance des objets. Nous avons aussi montré que l'algorithme [RB09] est extrêmement dépendant des étiquettes. En effet nous ne sommes pas arrivé à obtenir des résultats satisfaisants sans elles. Ces méthodes permettent de comparer n'importe quel objet modélisable avec ces structures de façon complètement générique. Elles restent dans la plupart des cas toujours moins performantes que des méthodes spécialisées. Par exemple, pour effectuer de la classification d'images, les méthodes de l'état de l'art utilisent des patchs, ou des points d'intérêts (SIFT) qui sont plus résistants aux bruits. Il conviendrait peut-être de chercher une méthode hybride pour laquelle la structure et les étiquettes sont considérées au même niveau.

Une autre distance d'édition Parallelement à nos travaux, [WDN+12] a récemment proposé une autre définition de la distance d'édition pour les cartes combinatoires 2D étiquetées. Ils proposent deux algorithmes, un premier basé sur l'algorithme $A*$, garantissant de trouver la solution optimale en temps exponentiel et un second de type glouton calculant une approximation de leur distance d'édition. Dans leur phase expérimentale, ils étudient les temps d'exécutions de leurs algorithmes, nous notons que le temps de calcul pour l'algorithme $A*$ est de 56s pour des cartes de 30 brins. Nous n'avons par contre pas d'information sur le temps d'exécution de l'algorithme glouton pour des tailles de cartes raisonables. Les autres expérimentations montrent que leur algorithme glouton est capable de classifier correctement des images segmentées. Cette publication étant très récente nous n'avons pas eu le temps de nous comparer autant au niveau expérimental que théorique.

Perspectives

Les cartes généralisées étant définies en toutes dimensions, il serait intéressant d'étudier le comportement de cette distance ainsi que des heuristiques proposées sur des jeux de données de dimensions supérieures à 3. Le principal problème étant le manque de jeux de données modélisables avec des cartes de dimensions supérieures à 3 pour lesquels nous avons une vérité terrain.

Nous souhaitons aussi étudier différentes méthodes pour modéliser des objets au moyen de cartes généralisées. Nous avons vu qu'il est possible de modéliser un objet de différentes manières avec la même structure de données. Une piste serait d'étudier

différentes manières de représenter un objet par une carte généralisée de sorte que le maximum d'information soit intégré dans la structure.

Nous avons proposé un algorithme calculant des solutions dirigées par des heuristiques locales. De nouvelles heuristiques globales permettant de choisir plusieurs points de départs simultanément ppourraient améliorer les résultats. Enfin, il serait aussi intéressant de réfléchir à un nouvel algorithme plus efficace en terme de complexité mémoire pour manipuler des cartes de plus grande taille. Nous avons remarqué que les méthodes statistiques obtiennent de très bons résultats sur des problèmes de classification. Exploiter les caractéristiques des cartes généralisées au sein de méthodes statistiques pourrait apporter de l'information sur la structure de l'objet modélisé.Enfin, nous souhaiterions aussi étudier les nouvelles mesures de similarités dites méthodes à noyaux sur les cartes généralisées en utilisant les informations qu'elles apportent.

Bibliographie

[AIM] Aimashape, http ://www.aimatshape.net/event/shrec/.

[AS64] M. Abramowitz and I.A. Stegun. *Handbook of Mathematical Functions*. Dover publications, 1964.

[Baa00] Martin Baatz. Multiresolution segmentation : an optimization approach for high quality multi-scale image segmentation. *Journal of Photogrammetry and Remote Sensing*, 58(3-4) :12–23, 2000.

[Bau72] Bruce G. Baumgart. Winged edge polyhedron representation. October 1972.

[BBG95] S.C. Basak, S. Bertelsen, and G.D. Grunwald. Use of graph theoretic parameters in risk assessment of chemicals. *Toxicology Letters*, 79(1–3) :239 – 250, 1995. <ce :title>Decision Subtances Methodologies for Human Health Risk Assessment of Toxic Substances</ce :title>.

[BFSS01a] C. Banderier, P. Flajolet, G. Schaeffer, and M. Soria. Random maps, coalescing saddles, singularity analysis, and Airy phenomena. *Random Structures Algorithms*, 19(3-4) :194–246, 2001. Analysis of algorithms (Krynica Morska, 2000).

[BFSS01b] C. Banderier, P. Flajolet, G. Schaeffer, and M. Soria. Random maps, coalescing saddles, singularity analysis, and Airy phenomena. *Random Structures Algorithms*, 19(3-4) :194–246, 2001. Analysis of algorithms (Krynica Morska, 2000).

[BGV92] Bernhard E. Boser, Isabelle Guyon, and Vladimir Vapnik. A training algorithm for optimal margin classifiers. In Haussler [Hau92], pages 144–152.

[BLV02] Achille J.-P. Braquelaire, Jacques-Olivier Lachaud, and Anne Vialard, editors. *Discrete Geometry for Computer Imagery, 10th International Conference, DGCI 2002, Bordeaux, France, April 3-5, 2002, Proceedings*, volume 2301 of *Lecture Notes in Computer Science*. Springer, 2002.

[BOS+05] Karsten M. Borgwardt, Cheng Soon Ong, Stefan Schönauer, S. V. N. Vishwanathan, Alex J. Smola, and Hans-Peter Kriegel. Protein function prediction via graph kernels, 2005.

[BR11] Horst Bunke and Kaspar Riesen. Recent advances in graph-based pattern recognition with applications in document analysis. *Pattern Recogn.*, 44 :1057–1067, May 2011.

[BRB04] Maria C. Boeres, Celso C. Ribeiro, and Isabelle Bloch. A randomized heuristic for scene recognition by graph matching. In Celso C. Ribeiro and

Simone L. Martins, editors, *WEA*, volume 3059 of *Lecture Notes in Computer Science*, pages 100–113. Springer, 2004.

[BRP09] Srecko Brlek, Christophe Reutenauer, and Xavier Provençal, editors. *Discrete Geometry for Computer Imagery, 15th IAPR International Conference, DGCI 2009, Montréal, Canada, September 30 - October 2, 2009. Proceedings*, volume 5810 of *Lecture Notes in Computer Science*. Springer, 2009.

[BRS10] O. Bodini, O. Roussel, and M. Soria. Boltzmann samplers for first order combinatorial differential equations. *Submitted*, 2010.

[BS98] H. Bunke and K. Shearer. A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters*, 19(3-4) :255–259, 1998.

[Bun97] H. Bunke. On a relation between graph edit distance and maximum common subgraph. *Pattern Recognition Letters*, 18 :689–694, August 1997.

[BV05] Luc Brun and Mario Vento, editors. *Graph-Based Representations in Pattern Recognition, 5th IAPR InternationalWorkshop, GbRPR 2005, Poitiers, France, April 11-13, 2005, Proceedings*, volume 3434 of *Lecture Notes in Computer Science*. Springer, 2005.

[BY86] Egon Balas and Chang Sung Yu. Finding a maximum clique in an arbitrary graph. *SIAM J. Comput.*, 15(4) :1054–1068, November 1986.

[CDS11] Camille Combier, Guillaume Damiand, and Christine Solnon. Measuring the distance of generalized maps. In Springer, editor, *Workshop on Graph-based Representations in Pattern Recognition*, Lecture Notes in Computer Science, pages 82–91. Springer Berlin/Heidelberg, May 2011.

[CFSV04] Luigi P. Cordella, Pasquale Foggia, Carlo Sansone, and Mario Vento. A (sub)graph isomorphism algorithm for matching large graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(10) :1367–1372, 2004.

[Chu97] Fan R K Chung. *Spectral Graph Theory*, volume 92. American Mathematical Society, 1997.

[CHV99] Olivier Chapelle, Patrick Haffner, and Vladimir Vapnik. Support vector machines for histogram-based image classification. *IEEE Transactions on Neural Networks*, 10(5) :1055–1064, 1999.

[CIE94] CIE. A method of predicting corresponding colours under different chromatic and illuminance adaptations. Technical report, Bureau central de la CIE, 1994.

[Cor73] Robert Cori. *Un code pour les graphes planaires et ses applications*. PhD thesis, 1973.

[CS03] Pierre-Antoine Champin and Christine Solnon. Measuring the similarity of labeled graphs. In *Proceedings of the 5th international conference on Case-based reasoning : Research and Development*, ICCBR'03, pages 80–95, Berlin, Heidelberg, 2003. Springer-Verlag.

[CSTD08] David Coeurjolly, Isabelle Sivignon, Laure Tougne, and Florent Dupont, editors. *Discrete Geometry for Computer Imagery, 14th IAPR International Conference, DGCI 2008, Lyon, France, April 16-18, 2008. Proceedings*, volume 4992 of *Lecture Notes in Computer Science*. Springer, 2008.

[Dam08] G. Damiand. Topological model for 3d image representation : Definition and incremental extraction algorithm. *Computer Vision and Image Understanding*, 109(3) :260–289, 2008.

[DBF04] G. Damiand, Y. Bertrand, and C. Fiorio. Topological model for two-dimensional image representation : definition and optimal extraction algorithm. *Computer Vision and Image Understanding*, 93(2) :111–154, February 2004.

[DD08] Alexandre Dupas and Guillaume Damiand. First results for 3d image segmentation with topological map. In Coeurjolly et al. [CSTD08], pages 507–518.

[DDL09] Alexandre Dupas, Guillaume Damiand, and Jacques-Olivier Lachaud. Multi-label simple points definition for 3d images digital deformable model. In Brlek et al. [BRP09], pages 156–167.

[DDLHJ+09] G. Damiand, C. De La Higuera, J.-C. Janodet, E. Samuel, and C. Solnon. Polynomial algorithm for submap isomorphism : Application to searching patterns in images. In *Workshop on Graph-based Representations in Pattern Recognition*, volume 5534 of *LNCS*, pages 102–112, Venice, Italy, May 2009. Springer Berlin/Heidelberg.

[DFLS04] P. Duchon, P. Flajolet, G. Louchard, and G. Schaeffer. Boltzmann samplers for the random generation of combinatorial structures. *Combinatorics, Probability and Computing*, 13 :2004, 2004.

[DL03] Guillaume Damiand and Pascal Lienhardt. Removal and contraction for n-dimensional generalized maps. In Nyström et al. [NdBS03], pages 408–419.

[DLHJS+] C. De La Higuera, J.-C. Janodet, E. Samuel, G. Damiand, and C. Solnon. Polynomial algorithms for open planar graph and subgraph isomorphism. Technical report.

[DP99] Pasari R. Durand P. An efficient algorithm for similarity analysis of molecules. *Internet Journal of Chemistry*, 1999.

[DPRS] A. Darrasse, K. Panagiotou, O. Roussel, and M. Soria. Boltzmann generation for regular languages with shuffle. *in preparation for Gascom'10, Montreal, Canada.*

[DR02] Guillaume Damiand and Patrick Resch. Topological map based algorithms for 3d image segmentation. In Braquelaire et al. [BLV02], pages 220–231.

[DSd+11] G. Damiand, C. Solnon, C. de la Higuera, J.-C. Janodet, and E. Samuel. Polynomial algorithms for subisomorphism of nd open combinatorial maps. *Computer Vision and Image Understanding (CVIU)*, 115(7) :996–1010, July 2011.

[Edm60] J. Edmonds. A combinatorial representation for polyhedral surfaces. *Notices Amer. Math. Soc.*, 7, 1960.

[Epp95] David Eppstein. Subgraph isomorphism in planar graphs and related problems. In *Proceedings of the sixth annual ACM-SIAM symposium on Discrete algorithms*, SODA '95, pages 632–640, Philadelphia, PA, USA, 1995. Society for Industrial and Applied Mathematics.

[FFP07] P. Flajolet, É. Fusy, and C. Pivoteau. Boltzmann sampling of unlabelled structures. In David Appelgate, editor, *Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments and the Fourth Workshop on Analytic Algorithmics and Combinatorics*, pages 201–211. SIAM Press, 2007. Proceedings of the New Orleans Conference.

[FLG00] Gary William Flake, Steve Lawrence, and C. Lee Giles. Efficient identification of web communities. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '00, pages 150–160, New York, NY, USA, 2000. ACM.

[FM03] Tom Fawcett and Nina Mishra, editors. *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*. AAAI Press, 2003.

[FS09] P. Flajolet and R. Sedgewick. *Analytic combinatorics*. Cambridge University Press, Cambridge, 2009.

[FV01] Mirtha-Lina Fernández and Gabriel Valiente. A graph distance metric combining maximum common subgraph and minimum common supergraph. *Pattern Recogn. Lett.*, 22 :753–758, May 2001.

[GBV11] Benoit Gauzere, Luc Brun, and Didier Villemin. Two new graph kernels and applications in chemoinformatics. In Xiaoyi Jiang, Miquel Ferrer, and Andrea Torsello, editors, *In 8th IAPR - TC-15 Workshop on Graph-based Representations in Pattern Recognition (GBR'11)*, volume 6658 of *Lecture Notes in Computer Science*, pages 112–122. Springer, May 2011.

[GDS11] Stéphane Gosselin, Guillaume Damiand, and Christine Solnon. Frequent Submap Discovery . In *22nd Annual Symposium on Combinatorial Pattern Matching (CPM2011)*, LNCS, pages 429–440, June 2011.

[GFW03] Thomas Gärtner, Peter Flach, and Stefan Wrobel. On graph kernels : Hardness results and efficient alternatives. In Bernhard Schölkopf and Manfred Warmuth, editors, *Learning Theory and Kernel Machines*, volume 2777 of *Lecture Notes in Computer Science*, pages 129–143. Springer Berlin / Heidelberg, 2003. 10.1007/978-3-540-45167-9_11.

[GJ79] M.R. Garey and D.S. Johnson. *Computers and Intractability : A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[Gos11] Stéphane Gosselin. *Recherche de motifs fréquents dans une base de cartes combinatoires*. Thèse de doctorat en informatique, Université Claude Bernard Lyon 1, November 2011.

[GPV00] Gaston H. Gonnet, Daniel Panario, and Alfredo Viola, editors. *LATIN 2000 : Theoretical Informatics, 4th Latin American Symposium, Punta del Este, Uruguay, April 10-14, 2000, Proceedings*, volume 1776 of *Lecture Notes in Computer Science*. Springer, 2000.

[GS85] Leonidas Guibas and Jorge Stolfi. Primitives for the manipulation of general subdivisions and the computation of Voronoi. *ACM Transactions on Graphics*, 4(2) :74–123, April 1985.

[HAA02] H.Kashima, A.Inokuchi, and A.Inokuchi. Kernels for graph classification. 2002.

[Hau92] David Haussler, editor. *Proceedings of the Fifth Annual ACM Conference on Computational Learning Theory, COLT 1992, Pittsburgh, PA, USA, July 27-29, 1992*. ACM, 1992.

[Hau99] David Haussler. Convolution kernels on discrete structures. Technical report, 1999.

[HDMB07] S. Horna, G. Damiand, D. Meneveaux, and Y. Bertrand. Building 3d indoor scenes topology from 2d architectural plans. In *Proc. of 2nd International Conference on Computer Graphics Theory and Applications (GRAPP)*, pages 37–44, Barcelona, Spain, March 2007.

[HMDB09] S. Horna, D. Meneveaux, G. Damiand, and Y. Bertrand. Consistency constraints and 3d building reconstruction. *Computer-Aided Design (CAD)*, 41(1) :13–27, January 2009.

[HW74] J. E. Hopcroft and J. K. Wong. Linear time algorithm for isomorphism of planar graphs (preliminary report). In *Proceedings of the sixth annual ACM symposium on Theory of computing*, STOC '74, pages 172–184, New York, NY, USA, 1974. ACM.

[JH06] Derek Justice and Alfred Hero. A binary linear programming formulation of the graph edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8) :1200–1214, 2006.

[Kao08] Ming-Yang Kao, editor. *Encyclopedia of Algorithms*. Springer, 2008.

[KL02] Risi Imre Kondor and John D. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *Proceedings of the Nineteenth International Conference on Machine Learning*, ICML '02, pages 315–322, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[Knu97] D. E. Knuth. *The Art of Computer Programming : Fundamental Algorithms*, volume 1. Addison Wesley, Reading, Massachusetts, 3rd edition, July 1997.

[KRGT10] Dakshina Ranjan Kisku, Ajita Rattani, Enrico Grosso, and Massimo Tistarelli. Face identification by sift-based complete graph topology. *2007 IEEE Workshop on Automatic Identification Advanced Technologies*, page 6, 2010.

[KStC03] Jaz Kandola, John Shawe-taylor, and Nello Cristianini. Learning semantic similarity. In *In NIPS*, pages 657–664. MIT Press, 2003.

[KTI03] Hisashi Kashima, Koji Tsuda, and Akihiro Inokuchi. Marginalized kernels between labeled graphs. In Fawcett and Mishra [FM03], pages 321–328.

[Lee11] Jimmy Ho-Man Lee, editor. *Principles and Practice of Constraint Programming - CP 2011 - 17th International Conference, CP 2011, Perugia, Italy, September 12-16, 2011. Proceedings*, volume 6876 of *Lecture Notes in Computer Science*. Springer, 2011.

[Lev65] V I Levenshtein. Binary codes capable of correcting spurious insertions and deletions of ones. *Problems of Information Transmission*, 1(1) :8–17, 1965.

[Lie91] Pascal Lienhardt. Topological models for boundary representation : a comparison with n-dimensional generalized maps. *Computer-Aided Design*, 23(1) :59–82, 1991.

[Lie94] P. Lienhardt. N-dimensional generalized combinatorial maps and cellular quasi-manifolds. *International Journal of Computational Geometry and Applications*, 4(3) :275–324, 1994.

[LK92] A. S. Lewis and G. Knowles. Image compression using the 2-D wavelet transform. *Image Processing, IEEE Transactions on*, 1(2) :244–250, April 1992.

[LMV01] Josep Lladoós, Enric Martí, and Juan Josée Villanueva. Symbol recognition by error-tolerant subgraph matching between region adjacency graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23 :1137–1143, October 2001.

[LSG+12] B. Li, Tobias Schreck, Afzal Godil, Marc Alexa, Tamy Boubekeur, Benjamin Bustos, J. Chen, Mathias Eitz, Takahiko Furuya, Kristian Hildebrand, S. Huang, H. Johan, Arjan Kuijper, Ryutarou Ohbuchi, Ronald Richter, Jose M. Saavedra, Maximilian Scherer, Tomohiro Yanagimachi, G. J. Yoon, and Sang Min Yoon. Shrec'12 track : Sketch-based 3d shape retrieval. In Spagnuolo et al. [SBBF12], pages 109–118.

[LWH03] Bin Luo, Richard C. Wilson, and Edwin R. Hancock. Spectral embedding of graphs. *Pattern Recognition*, 36(10) :2213 – 2230, 2003.

[LWHH03] Bin Luo, Richard C. Wilson, Edwin R. Hancock, and Edwin R. Hancock. Spectral embedding of graphs. pages 2213–2230, 2003.

[Mal89] Stephane G. Mallat. A theory for multiresolution signal decomposition : the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11 :674–693, 1989.

[Man88] M. Mantyla. Introduction to Solid Modeling. January 1988.

[McG82] James J. McGregor. Backtrack search algorithms and the maximal common subgraph problem. *Software Practice and Experience*, 12(1) :23–34, 1982.

[McK08] Brendan D. McKay. Graph isomorphism. In Kao [Kao08].

[Mer09] J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philos. Trans. Royal Soc. (A)*, 83(559) :69–70, November 1909.

[Mez05] Paul G. Mezey. Graph representations of molecular similarity measures based on topological resolution. *J. Comp. Methods in Sci. and Eng.*, 5 :109–114, January 2005.

[MOK] Moka, 3d topological modeler.

[MP08] T. Milenkovic and N. Przulj. Uncovering Biological Network Function via Graphlet Degree Signatures. *arXiv*, q-bio.MN, January 2008.

[Mun57] James R. Munkres. Algorithms for the Assignment and Transportation Problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1) :32–38, March 1957.

[NdBS03] Ingela Nyström, Gabriella Sanniti di Baja, and Stina Svensson, editors. *Discrete Geometry for Computer Imagery, 11th International Conference, DGCI 2003, Naples, Italy, November 19-21, 2003, Proceedings*, volume 2886 of *Lecture Notes in Computer Science*. Springer, 2003.

[Nil82] Nils J. Nilsson. *Principles of Artificial Intelligence*. Springer, 1982.

[NS11] Samba Ndojh Ndiaye and Christine Solnon. Cp models for maximum common subgraph problems. In Lee [Lee11], pages 637–644.

[OPR78] Ron Ohlander, Keith Price, and D. Raj Reddy. Picture segmentation using a recursive region splitting method. *Computer Graphics and Image Processing*, 8(3) :313 – 333, 1978.

[PB61] W W Peterson and D T Brown. *Cyclic Codes for Error Detection*, volume 49, pages 228–235. 1961.

[PBMW99] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking : Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

[PCFSV04] Luigi P. Cordella, Pasquale Foggia, Carlo Sansone, and Mario Vento. A (sub)graph isomorphism algorithm for matching large graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(10) :1367–1372, October 2004.

[PD05] Elzbieta Pekalska and Robert P. W. Duin. *The Dissimilarity Representation for Pattern Recognition : Foundations And Applications (Machine Perception and Artificial Intelligence)*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2005.

[R02] John W. Raymond, Peter Willett 0002, and Peter Willett 0002. Maximum common subgraph isomorphism algorithms for the matching of chemical structures. pages 521–533, 2002.

[Rag00] Prabhakar Raghavan. Graph structure of the web : A survey. In Gonnet et al. [GPV00], pages 123–125.

[RB09] Kaspar Riesen and Horst Bunke. Approximate graph edit distance computation by means of bipartite graph matching. *Image Vision Comput.*, 27 :950–959, June 2009.

[RF81] D.F. Robinson and L.R. Foulds. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53(1–2) :131 – 147, 1981.

[RGW02] J W Raymond, E J Gardiner, and P Willett. Rascal : calculation of graph similarity using maximum common edge subgraphs. *The Computer Journal*, 45(6) :631–644, 2002.

[RKH07] Antonio Robles-Kelly and Edwin R. Hancock. A riemannian approach to graph embedding. *Pattern Recogn.*, 40 :1042–1056, March 2007.

[RS09] O. Roussel and M. Soria. Boltzmann sampling of ordered structures. *Electronic Notes in Discrete Mathematics*, 35 :305–310, 2009. LAGOS'09 - V Latin-American Algorithms, Graphs and Optimization Symposium.

[SBBF12] Michela Spagnuolo, Michael M. Bronstein, Alexander M. Bronstein, and Alfredo Ferreira, editors. *Eurographics Workshop on 3D Object Retrieval 2012, Cagliari, Italy, May 13, 2012. Proceedings*. Eurographics Association, 2012.

[SDL06] C. Simon, G. Damiand, and P. Lienhardt. nd generalized map pyramids : Definition, representations and basic operations. *Pattern Recognition (PR)*, 39(4) :527–538, April 2006.

[SFTA08] Akira Suga, Keita Fukuda, Tetsuya Takiguchi, and Yasuo Ariki. Object recognition and segmentation using sift and graph cuts. *2008 19th International Conference on Pattern Recognition*, pages 1–4, 2008.

[SHB07] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image Processing, Analysis, and Machine Vision*. Thomson-Engineering, 2007.

[SJ89] B. W. Silverman and M. C. Jones. E. Fix and J.L. Hodges (1951) : An Important Contribution to Nonparametric Discriminant Analysis and Density Estimation : Commentary on Fix and Hodges (1951). *International Statistical Review / Revue Internationale de Statistique*, 57(3) :233–238, 1989.

[SJ07] Christine Solnon and Jean-Michel Jolion. Generalized vs set median strings for histogram-based distances : algorithms and classification results in the image domain. In *Proceedings of the 6th IAPR-TC-15 international conference on Graph-based representations in pattern recognition*, GbRPR'07, pages 404–414, Berlin, Heidelberg, 2007. Springer-Verlag.

[SNB+06] Barbara Spillmann, Michel Neuhaus, Horst Bunke, Elżbieta Pękalska, and Robert P. W. Duin. Transforming strings to vector spaces using prototype selection. In *Proceedings of the 2006 joint IAPR international conference on Structural, Syntactic, and Statistical Pattern Recognition*, SSPR'06/SPR'06, pages 287–296, Berlin, Heidelberg, 2006. Springer-Verlag.

[Sol10] Christine Solnon. Alldifferent-based filtering for subgraph isomorphism. *Artif. Intell.*, 174(12-13) :850–864, 2010.

[SS04] Sébastien Sorlin and Christine Solnon. Reactive Tabu Search for Measuring Graph Similarity. Technical Report RR-LIRIS-2004-035, LIRIS UMR 5205 CNRS/INSA de Lyon/Université Claude Bernard Lyon 1/Université Lumière Lyon 2/École Centrale de Lyon, November 2004.

[SS05] Sébastien Sorlin and Christine Solnon. Reactive tabu search for measuring graph similarity. In Brun and Vento [BV05], pages 172–182.

[SSG05] Olfa Sammoud, Christine Solnon, and Khaled Ghédira. Ant algorithm for the graph matching problem. In Günther Raidl and Jens Gottlieb, editors, *Evolutionary Computation in Combinatorial Optimization*, volume 3448 of *Lecture Notes in Computer Science*, pages 213–223. Springer Berlin / Heidelberg, 2005. 10.1007/978-3-540-31996-2.

[SSSG06] Olfa Sammoud, Sébastien Sorlin, Christine Solnon, and Khaled Ghédira. A comparative study of ant colony optimization and reactive search for graph matching problems. In *Proceedings of the 6th European conference on Evolutionary Computation in Combinatorial Optimization*, EvoCOP'06, pages 234–246, Berlin, Heidelberg, 2006. Springer-Verlag.

[SVP+09] Nino Shervashidze, S. V. N. Vishwanathan, Tobias Petri, Kurt Mehlhorn, Karsten M. Borgwardt, and Karsten M. Borgwardt. Efficient graphlet kernels for large graph comparison. pages 488–495, 2009.

[SWD] Gaurav Sharma, Wencheng Wu, and Edul N. Dalal. The CIEDE2000 color-difference formula : Implementation notes, supplementary test data, and mathematical observations. *Color Research & Application*, 30(1) :21–30, February.

[Sys82] Maciej M Sysło. The subgraph isomorphism problem for outerplanar graphs. *Theoretical Computer Science*, 17(1) :91–97, 1982.

[Tor] Warren S. Torgerson. *Theory and methods of scaling*. R.E. Krieger Pub. Co.

[TSS92] Yoshimasa Takahashi, Masayuki Sukekawa, and Shinichi Sasaki. Automatic identification of molecular similarity using reduced-graph representation of chemical structure. *Journal of Chemical Information and Computer Sciences*, 32(6) :639–643, 1992.

[Ull76] J. R. Ullmann. An algorithm for subgraph isomorphism. *J. ACM*, 23(1) :31–42, January 1976.

[Vap98] Vladimir Vapnik. *Statistical learning theory*. Wiley, 1998.

[VNU03] Michel Vidal-Naquet and Shimon Ullman. Object recognition with informative features and linear classification. In *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, ICCV '03, pages 281–, Washington, DC, USA, 2003. IEEE Computer Society.

[Wat99] Chris Watkins. Dynamic alignment kernels. In *Advances in Large Margin Classifiers*, pages 39–50. MIT Press, 1999.

[WDN+12] Tao Wang, Guojun Dai, Bingbing Ni, De Xu, and Francois Siewe. A distance measure between labeled combinatorial maps. *Computer Vision and Image Understanding*, (0) :–, 2012.

[ZAM85] Rosenfeld, a. (ed.), multiresolution image processing and analysis. berlin-heidelberg-new york-tokyo, springer-verlag 1984. viii, 385 s., 198 abb., dm 75,—. us $ 28.00. isbn 3-540-13006-3 (springer-series in information sciences 12). *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 65(7) :274–274, 1985.

[ZDS10] Stéphane Zampelli, Yves Deville, and Christine Solnon. Solving subgraph isomorphism problems with constraint programming. *Constraints*, 15(3) :327–353, 2010.