

Université de Bordeaux I
Laboratoire Bordelais de Recherche en Informatique

HABILITATION À DIRIGER DES RECHERCHES

au titre de l'école doctorale de Mathématiques et Informatique de Bordeaux

soutenue et présentée publiquement le 3 avril 2013

par Monsieur Nicolas BONICHON

Quelques algorithmes entre le monde des graphes et les nuages de points.

Rapporteurs

Monsieur Prosenjit Bose	Professeur à l'université de Carleton
Monsieur Christophe Paul	Directeur de recherches au LIRMM
Monsieur Laurent Viennot	Directeur de recherches au LIAFA

Membres du Jury

Monsieur Olivier Beaumont	Directeur de recherches chez Inria Bordeaux Sud-Ouest
Monsieur Robert Cori	Professeur émérite à l'université de Bordeaux
Monsieur Olivier Devillers	Directeur de recherches chez Inria Sophia-Antipolis
Monsieur Philippe Duchon	Professeur à l'université Bordeaux 1
Monsieur Christophe Paul	Directeur de recherches au LIRMM
Monsieur Laurent Viennot	Directeur de recherches au LIAFA

Table des matières

1	Combinatoire Bijective : Orientations bipolaires et permutations de Baxter	3
1.1	Introduction	4
1.2	Préliminaires	5
1.2.1	Permutations de Baxter	5
1.2.2	Orientations bipolaires planes	6
1.3	Présentation de la bijection	7
1.3.1	Des permutations de Baxter aux orientations bipolaires planes	7
1.3.2	La bijection inverse	8
1.3.3	Symétries de la bijection	8
1.4	Spécialisations	10
1.4.1	Permutations de Baxter évitant le motif 2413 et les cartes enracinées non-séparables	10
1.4.2	Permutations évitant les motifs 2413 et 3142 et les cartes série-parallèles	12
1.5	Digression sur les spécialisations	13
1.6	Conclusion et perspectives	14
2	Dessin de graphes : Dessin convexe de cartes planaires 3-connexes	17
2.1	Introduction	18
2.2	Bois de Schnyder et algorithme par comptage de faces	18
2.2.1	Définitions	18
2.2.2	Dessin convexe par comptage de faces	20
2.2.3	Treillis des bois de Schnyder	21
2.2.4	Fusion et Division	23
2.3	L'algorithme de dessin convexe compact	24
2.3.1	Le dessin est convexe	25
2.3.2	Nombre de fusions	26
2.4	Améliorations et Limitations	29
2.4.1	Etude en moyenne	30
2.5	Conclusion et Perspectives	30
3	Graphes couvrants géométriques	33
3.1	Introduction	34
3.2	Unification	36
3.2.1	Demi Θ_6 -graphe	36
3.2.2	Plongement géodésique	36
3.2.3	Triangulation de TD-Delaunay	38
3.3	Unification des concepts	39
3.4	Applications	40
3.4.1	Spanner	40
3.4.2	Réalisabilité de Delaunay	40
3.4.3	Algorithme de construction	40
3.5	Spanner de degré borné	40
3.6	Un 6-Spanner simple planaire de degré maximum 9	41

3.7	Un 6-Spanner planaire de degré maximum 6	44
3.8	Conclusion et perspectives	46
4	Algorithmes d'approximation : Agrégation de ressources hétérogènes	53
4.1	Introduction	54
4.2	Définitions et résultats d'inapproximation	55
4.2.1	Définitions et notations	55
4.2.2	Inapproximabilité avec faible augmentation de ressources	56
4.2.2.1	Inapproximabilité de BCCD	56
4.2.2.2	Inapproximabilité de BPCD	57
4.3	BCCD : Algorithmes d'approximations avec augmentation de ressources	58
4.3.1	Algorithme d'approximation pour des ρ -inframétriques	58
4.3.2	$(\frac{2}{5}, 2 + \frac{3\sqrt{3}}{2})$ -algorithme d'approximation de BCCD pour un espace de Vivaldi	59
4.3.3	Métrique d'arbre	61
4.4	Algorithmes d'approximations pour BPCD	61
4.5	Vers une utilisation dans les plateformes à grande échelle	62
4.5.1	Latence et Outils de plongements	63
4.5.2	Évaluation expérimentale	63
4.6	Conclusion et perspectives	65

Introduction

Ce document retrace une partie de mes activités de recherches réalisées après l’obtention de mon doctorat en décembre 2002. Ces recherches ont été menées au sein de l’équipe *Combinatoire et Algorithmique* du LaBRI. Depuis 2007 ces recherches se font également au sein de l’équipe projet Inria *CEPAGE*.

En regardant en arrière je m’aperçois qu’une partie importante de mes recherches a consisté à concevoir des algorithmes qui, partant d’un graphe, calculent un nuage de points ou bien partant d’un nuage de points calculent un graphe. Cette formulation simpliste dissimule en réalité une grande diversité des problèmes traités et des techniques utilisées. Comme nous allons le voir par la suite ce “passe-temps” m’a permis d’apporter des contributions en combinatoire bijective (Chapitre 1), en tracé de graphes (Chapitre 2), en géométrie algorithmique (Chapitre 3) et en complexité (Chapitre 4). La suite du document est organisée comme indiqué ci-dessous.

Dans le Chapitre 1, nous nous intéressons aux orientations bipolaires planes (qui sont des cartes planaires décorées d’une orientation particulière) et aux nuages de points correspondants aux diagrammes de permutations de Baxter. Nous présentons une bijection (ainsi que son inverse) transformant les orientations bipolaires planes (à n arêtes) en permutations de Baxter (de taille n). La bijection présentée a la propriété recherchée d’envoyer de nombreux paramètres classiques sur les permutations vers des paramètres tout aussi naturels sur les cartes planaires.

Dans le Chapitre 2, nous présentons un algorithme qui, partant d’une carte plane 3-connexe, plonge les sommets de cette carte sur une grille (de petite taille) de sorte que le dessin en ligne droite du graphe induit par ce plongement soit sans croisement convexe (*i.e.* les faces sont représentées par des polygones convexes).

Dans le Chapitre 3, nous effectuons le voyage inverse : partant d’un nuage de points du plan, nous cherchons à calculer un *spanner*, c’est à dire graphe plane s’appuyant sur ces points de sorte que l’étirement du graphe (*i.e.* le rapport entre la distance dans ce graphe et la distance Eulérienne) soit petit. Nous montrons en particulier que les graphes Θ_6 ont un étirement d’au plus 2 (on ignorait avant si leur étirement était borné) et qu’il est possible de construire des spanners planaires de degré 6 et d’étirement au plus 6, améliorant ainsi significativement la meilleure borne connue sur le degré pour un spanner plane (anciennement 11).

Enfin, dans le Chapitre 4, nous nous intéressons à des problèmes d’agrégation de ressources : partant d’un nuage de points pondérés, nous cherchons à construire sur ces points des cliques disjointes de diamètre borné. Dans le premier problème que nous regardons (*bin covering avec contrainte de distance*) chaque clique doit avoir un poids d’au moins un. Dans le deuxième problème (*bin packing avec contrainte de distance*) chaque point doit appartenir à exactement une clique et chaque clique doit avoir un poids inférieur à 1. Pour ces 2 problèmes, nous montrons qu’il n’existe pas d’algorithme d’approximation (même en relâchant légèrement la contrainte sur le diamètre des cliques), sauf si $P = NP$. Par ailleurs, nous proposons des algorithmes d’approximation en relâchant raisonnablement la contrainte sur le diamètre des cliques. Les résultats présentés dans ce chapitre correspondent à une partie importante des travaux de thèse d’Hubert Larchevêque.

Chapitre 1

Combinatoire Bijective : Orientations bipolaires et permutations de Baxter

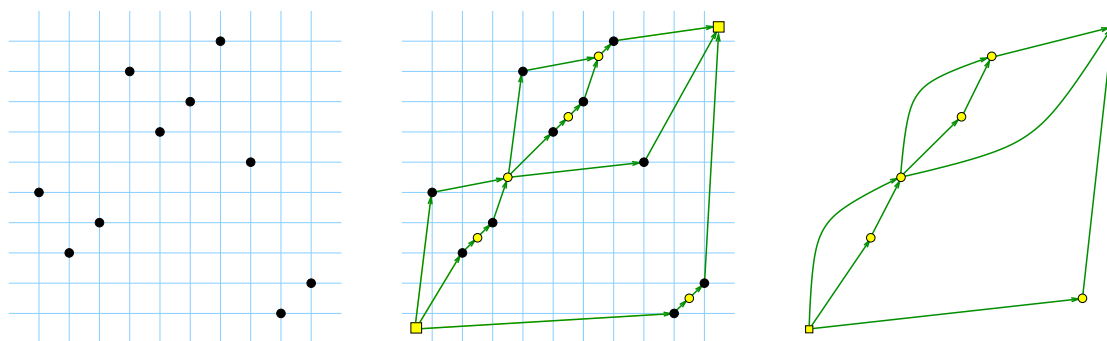


FIGURE 1.1 – La permutation de Baxter $\pi = 53497810612$, le diagramme de Hasse de $\phi(\pi)$ et l'orientation bipolaire plane $\Phi(\pi)$.

Les cartes, les arbres, les permutations et les mots sont parmi les objets favoris de la combinatoire. Ils le sont à double titre : d'abord parce qu'ils ont de nombreuses propriétés remarquables et ensuite parce qu'ils apparaissent de manière récurrente dans de nombreux domaines scientifiques (physique statistique, algèbre, algorithmique, bio-informatique). Un des buts de la combinatoire bijective est d'établir des bijections entre différentes familles d'objets. Ces bijections permettent de transposer les résultats connus dans chaque famille vers l'autre enrichissant ainsi chacune des familles de nouvelles propriétés.

Par exemple, le travail que nous avons réalisé en collaboration avec Cyril Gavoille et Nicolas Hanusse [BGH05] illustre bien ce propos. Nous avons proposé une bijection entre d'un côté les cartes planaires extérieures (*i.e.* dont tous les sommets sont sur la face extérieure) à n sommets et m arêtes et de l'autre les arbres bicoloriés à n sommets et $m - n$ sommets noirs (non situés sur la dernière branche). Cette bijection se spécialise également aux cartes planaires extérieures *2-connexes*. De cette bijection nous en déduisons 3 résultats sur les cartes planaires extérieures :

- une formule d'énumération avec une asymptotique en $3n^{O(\log n)}$;
- une structure de données de taille asymptotiquement optimale ($3n$ bits), calculée en temps linéaire (en outre cette structure de données permet de renvoyer l'adjacence entre 2 noeuds en temps constant et de déterminer les voisins d'un sommet de degré d en $O(d)$) ;
- un algorithme de génération aléatoire uniforme dont la complexité moyenne est linéaire.

Ce chapitre présente une bijection élégante (ainsi que ses implications) entre les permutations de Baxter de taille n et les orientations bipolaires planes à n arêtes. Ce travail a été réalisé en collaboration

avec Mireille Bousquet-Mélou et Éric Fusy (LIX - Palaiseau) [BBMF10]. Le travail présenté dans la Section 1.5 a été réalisé en collaboration avec Olivier Bernardi [BB09].

1.1 Introduction

Nous nous intéressons ici à une famille de permutation introduite par Glen Baxter en 1964 et qui porte son nom [Bax64]. Une permutation π est *de Baxter* s'il n'existe pas d'indices $i < j < k$ tels que $\pi(j+1) < \pi(i) < \pi(k) < \pi(j)$ ou $\pi(j) < \pi(k) < \pi(i) < \pi(j+1)$. Ces permutations ont été énumérées pour la première fois dans les années 80 [CGHK78, Mal79, Vie84, CDV86]. Plus récemment, elles ont été étudiées dans le contexte plus général des *permutations à motifs exclus* [BM03, DG96, Gir93, GL00]. En particulier le nombre de permutations de Baxter de \mathfrak{S}_n ayant m montées, i maxima à gauche¹, j maxima à droite est [Mal79] :

$$\frac{ij}{n(n+1)} \binom{n+1}{m+1} \left[\binom{n-i-1}{n-m-2} \binom{n-j-1}{m-1} - \binom{n-i-1}{n-m-1} \binom{n-j-1}{m} \right]. \quad (1.1)$$

Quelques années plus tard, un autre Baxter, Rodney Baxter, étudia la somme du polynôme de Tutte $T_M(x, y)$ sur l'ensemble des cartes planaires non-séparables M ayant une taille fixée [Bax01]. Il a montré que le coefficient de $x^1 y^0$ dans $T_M(x, y)$, sommé sur toutes les cartes planaires non-séparables enracinées M ayant $n+1$ arêtes, $m+2$ sommets, dont le degré de la face racine est $i+1$ et dont le degré du sommet racine est $j+1$, était donné par la même formule (1.1). En fait, le coefficient $x^1 y^0$ dans $T_M(x, y)$ correspond au nombre d'orientations bipolaires de M [GZ83, GS00].

Mireille Bousquet-Mélou [BM03] fut la première à remarquer que ces deux Baxter étudiaient des objets énumérés de la même manière.

Dans ce chapitre nous allons présenter une bijection directe entre les permutations de Baxter et les orientations bipolaires planes. Cette bijection montre clairement pourquoi les objets introduits par les deux Baxter sont énumérés par les mêmes nombres. Un des attraits de notre bijection est qu'elle est non-réursive, simple à implémenter et qu'elle a beaucoup de structure : elle traduit plusieurs statistiques naturelles des permutations sur les cartes. De plus elle se comporte bien vis-à-vis des symétries. Par ailleurs, lorsqu'on applique cette bijection aux permutations de Baxter évitant le motif 2413, cette bijection se spécialise en une bijection entre ces permutations et les cartes planaires enracinées non-séparables. Cette dernière bijection est étroitement liée à la description récursive de ces cartes introduite par Dulucq et al. [DGW96].

Les preuves de cette bijection reposent sur l'existence de deux arbres de générations isomorphes pour les permutations de Baxter et pour les orientations bipolaires planes.

Notons qu'il existe d'autres objets énumérés par les nombres de Baxter. Citons par exemple les *floorplan* [ABP06], les paires d'arbres jumeaux [DG96] ainsi qu'une famille de triplés de chemins ne se coupant pas [FPS09]. D'autres bijections sont présentées dans [FFNO]. Toutefois la bijection présentée ici ne peut s'exprimer comme la composition de bijections existantes et offre certaines propriétés intéressantes (comme les symétries, les spécialisations, ...).

A l'instar des objets comptés par les nombres de Catalan, les liens entre les objets comptés par les nombres de Baxter commencent à être mieux compris.

La suite de ce chapitre est organisée comme suit : après l'introduction de quelques définitions et notations dans la section 1.2, nous présentons dans la section 1.3 la bijection Φ , son inverse Φ^{-1} , ainsi que certaines propriétés de symétrie de Φ . La Section 1.4 présente des spécialisations de cette bijection aux cartes non-séparables et aux cartes séries-parallèles. La Section 1.5 développe quelques digressions sur l'intérêt de considérer des cartes enrichies pour découvrir de nouvelles bijections. Enfin une dernière section conclut et présente quelques perspectives.

1. On dit que i est une *montée* (resp. une *descente*) de π si $\pi(i) < \pi(i+1)$ (resp. $\pi(i) > \pi(i+1)$). i est un *maximum à gauche* (resp. à *droite*) si pour tout $j < i$ (resp. $j > i$) $\pi(i) > \pi(j)$.

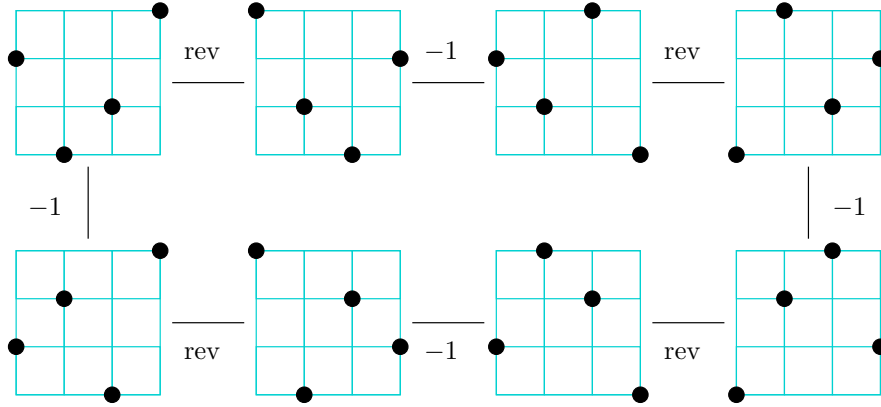


FIGURE 1.2 – Les symétries du carré agissant sur les permutations de Baxter.

1.2 Préliminaires

1.2.1 Permutations de Baxter

Une *permutation* $\pi = \pi(1)\dots\pi(n)$ est une application bijective qui va de l'ensemble $\{1, 2, \dots, n\}$ dans lui-même. On note \mathfrak{S}_n l'ensemble des permutations de taille n . Une permutation $\pi = \pi(1)\dots\pi(n)$ sera représenté par son *diagramme*, c'est à dire par l'ensemble de points $(i, \pi(i))$. Ainsi, le groupe des symétries du carré (d'ordre 8) agit sur les permutations de \mathfrak{S}_n . Ce groupe peut être généré par les 2 involutions suivantes :

- l'inversion $\pi \mapsto \pi^{-1}$, qui revient à faire une symétrie par rapport à la première diagonale,
- le retournement de permutation qui envoie $\pi = \pi(1)\dots\pi(n)$ sur $\text{rev}(\pi) = \pi(n)\dots\pi(1)$.

D'après la définition que nous avons donnée dans l'introduction des permutations de Baxter, il apparaît clairement que les opérations d'inversion et de retournement laissent l'ensemble \mathcal{B}_n des permutations de Baxter de taille n inchangé. Cela implique que les 8 symétries du carré (Figure 1.2) laissent également inchangé l'ensemble \mathcal{B}_n .

Les permutations de Baxter peuvent également être définies en termes de *motifs barrés* [Gir93], comme nous allons le faire maintenant.

Étant donné 2 permutations π et $\tau = \tau_1 \dots \tau_k$, une *occurrence* du motif τ dans π est une sous-séquence $\pi(i_1), \dots, \pi(i_k)$, avec $i_1 < \dots < i_k$, tel que pour tout couple $(j, l), \pi(i_j) < \pi(i_l)$ si et seulement si $\tau_j < \tau_l$. Si π ne contient aucune occurrence de τ , on dit que π *évite le motif* τ . On dit que π évite le motif barré $\tau = \tau_1 \dots \tau_{i-1} \bar{\tau}_i \tau_{i+1} \dots \tau_k$ si toute occurrence de $\tau_1 \dots \tau_{i-1} \tau_{i+1} \dots \tau_k$ dans π est une sous-séquence d'une occurrence de $\tau_1 \dots \tau_{i-1} \tau_i \tau_{i+1} \dots \tau_k$. Avec ces définitions, les permutations de Baxter sont exactement les permutations qui évitent les motifs $25\bar{3}14$ et $41\bar{3}52$.

Nous équipons maintenant \mathbb{R}^2 de l'ordre suivant : $v \leq w$ si $x(v) \leq x(w)$ et $y(v) \leq y(w)$. Cet ordre correspond à “ w est au nord-est de v ”. On dit que pour 2 éléments v et w d'un ensemble partiellement ordonné \mathcal{P} , w *couvre* v si $v < w$ et qu'il n'existe pas de élément u tel que $v < u < w$. Le *diagramme de Hasse* de \mathcal{P} est le graphe orienté dont \mathcal{P} est l'ensemble des sommets et les relations de couvertures correspondent aux arcs. Nous orientons les arcs du plus petit au plus grand.

Étant donnée une permutation π , on dit que i est une *montée* (resp. une *descente*) si $\pi(i+1) > \pi(i)$ (resp. $\pi(i+1) < \pi(i)$). Un *maximum à gauche* (ou *lr-maximum* pour “left to right” maximum) est une valeur $\pi(i)$ tel que $\pi(i) > \pi(j)$ pour tout $j < i$. On définit de manière analogue *rl-maximum*, *lr-minimum*, *rl-minimum*.

1.2.2 Orientations bipolaires planes

Une *carte plane* est un graphe connexe plongé sur la sphère sans croisement d'arêtes, considéré à déformation continue près. Une carte est composée de sommets, d'arêtes ainsi que de *faces* qui sont les composantes connexes de \mathbb{R}^2 après suppression des arêtes. On appelle *degré d'une face* le nombre de sommets qui bordent cette face. La *face externe* est non bornée et les *faces internes* sont bornées. Une carte est dite *séparable*, s'il existe un sommet séparateur, c'est-à-dire que sa suppression déconnecte le graphe. Une *orientation bipolaire plane* O est une orientation acyclique d'une carte plane M avec une unique *source* s (sommet sans arc entrant) et un unique *puits* t (sommet sans arc sortant), tous deux situés sur la face externe (Figure 1.3). Les *pôles* de O sont les sommets s et t . Un des chemins allant de la source au puits a la face externe à sa droite : ce chemin est le *bord droit* de O et sa longueur est appelé *degré sortant droit* de O . Le *degré sortant gauche* est défini de manière similaire.

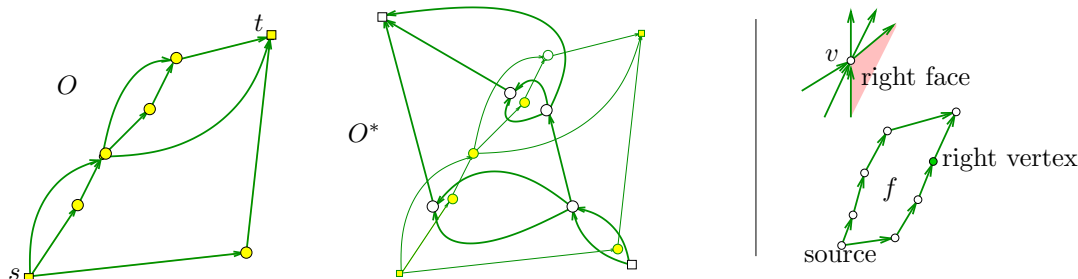


FIGURE 1.3 – A gauche, une orientation bipolaire plane ayant un degré sortant droit (gauche) de 2 (3) ainsi que son dual. A droite, une illustration des propriétés locales des orientations bipolaires planes.

On peut voir qu'autour de chaque sommet non-polaire v d'une orientation bipolaire plane, les arcs sont organisés en 2 blocs, un bloc d'arcs entrants et un bloc d'arcs sortants : en tournant autour de v on parcourt le bloc d'arcs entrants, puis le bloc d'arcs sortants et puis c'est tout (Figure 1.3, à droite). La face qui est incidente au dernier arc sortant et au premier arc entrant (dans le sens horaire) est appelée *face droite* de v . De manière similaire, la *face gauche* de v est incidente au dernier arc entrant et au premier arc sortant.

De manière duale, le bord de chaque face interne f contient exactement 2 chemins dirigés maximaux (Figure 1.3, à droite), formant une petite orientation bipolaire. Sa source (resp. puits) est appelé *source* (resp. *puits*) de f . Les autres sommets de la face sont respectivement appelés *sommets droits*, *sommets gauches* de f : si v est un sommet droit de f , alors f est la face gauche de v .

Une *carte* M est *enracinée*, si une des arêtes adjacentes à la face externe est orientée de sorte que la face externe se trouve à droite. Dans le cas des orientations bipolaires de M , la source s et le puits t doivent être les extrémités de l'arc racine. Comme rappelé dans l'introduction de ce chapitre, le nombre d'orientations bipolaires de M est égal au coefficient de x^1y^0 dans le polynôme de Tutte $T_M(x, y)$. Ce nombre est non nul si et seulement si M est non séparable [LEC67].

Une carte est *bipolaire*, si 2 de ses sommets de la face externe et respectivement appelés *source* (s) et *destination* (t) sont distingués. Dans ce cas, une orientation bipolaire de M doit avoir s comme source et t comme puits. Par la suite si M est enracinée, \tilde{M} est la carte bipolaire obtenue par suppression de l'arc racine. Si M est bipolaire, \hat{M} est la carte obtenue par ajout de l'arc racine. Dans ce cas, M admet une orientation bipolaire si et seulement si \hat{M} est non-séparable.

Deux transformations naturelles agissent sur l'ensemble \mathcal{O}_n des orientations bipolaires planes (non enracinées) ayant n arcs. Pour une orientation bipolaire O , nous définissons $\text{mir}(O)$ l'*image miroir* de O , c'est-à-dire l'orientation obtenue en retournant O par rapport à une droite quelconque (Figure 1.4). Clairement l'opération mir est une involution.

L'autre transformation est la *dualité*. Le dual d'une orientation plane O^* de O est construit comme indiqué dans la Figure 1.3. Il y a un sommet de O^* pour chaque face interne de O et 2 sommets de O^* (ses

pôles) pour la face externe de O . Il y a un arc dans O^* connectant les sommets correspondants à deux faces de O partageant un arc commun et orienté en utilisant la convention présentée dans la Figure 1.3 : de la face située à droite de l'arc vers la face située à gauche de l'arc. Remarquons, que le dual du dual de O est obtenu à partir de O en retournant tous les arcs. La transformation duale est donc d'ordre 4. Les transformations mir et $O \mapsto O^*$ engendrent un groupe d'ordre 8 (Figure 1.4). Nous allons voir que notre bijection Φ nous permet de superposer les Figures 1.2 et 1.4. C'est à dire, si $\Phi(\pi) = O$, alors $\Phi(\pi^{-1}) = \text{mir}(O)$ et $\Phi(\text{rev}(\pi)) = \text{mir}(O^*)$.

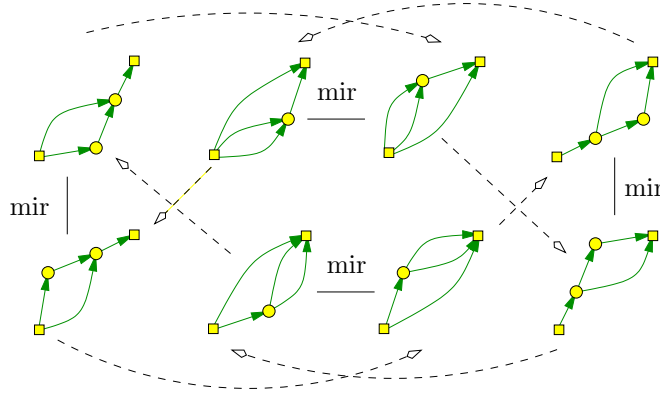


FIGURE 1.4 – Un groupe d'ordre 8 agissant sur les orientations bipolaires planes. Les arcs en pointillés indiquent le passage d'une orientation à son orientation duale, les autres arcs indiquent l'application de l'opération miroir.

1.3 Présentation de la bijection

1.3.1 Des permutations de Baxter aux orientations bipolaires planes

Soit π une permutation de Baxter. Construisons d'abord un graphe orienté plongé biparti $\phi(\pi)$ avec segments de droites pour arêtes. Les sommets noirs sont les points $b_i = (i, \pi(i))$. Les sommets blancs correspondent aux *montées* de π . Plus précisément, pour chaque montée a (i.e., $\pi(a) < \pi(a+1)$), notons $\ell_a = \max\{\pi(i) : i < a+1 \text{ et } \pi(i) < \pi(a+1)\}$. La propriété Baxter implique que pour tout $i > a$ tel que $\pi(i) > \pi(a)$, on a $\pi(i) > \ell_a$ (voir Figure 1.5). Le sommet blanc associé à la montée a est $w_a = (a+1/2, \ell_a+1/2)$. Finalement, nous ajoutons 2 sommets blancs $w_0 = (1/2, 1/2)$ et $w_n = (n+1/2, n+1/2)$. (En d'autres termes on considère que $\pi(0) = 0$ et $\pi(n+1) = n+1$.) Nous définissons le graphe orienté plongé $\phi(\pi)$ comme étant le diagramme de Hasse de l'ensemble des points noirs et blancs pour l'ordre "est au nord-est de" défini sur \mathbb{R}^2 dont les arcs sont représentés par des segments de droites (Figure 1.1, centre).

Proposition 1.1. *Pour chaque permutation de Baxter π , le graphe plongé $\phi(\pi)$ est planaire, bicolore (les extrémités de chaque arête sont de couleurs différentes) et le degré sortant et entrant de chaque sommet noir est de 1.*

Ces propriétés peuvent être observées sur l'exemple de la Figure 1.1.

En supprimant les sommets noirs, nous obtenons une orientation bipolaire plane ayant pour source w_0 et pour puits w_n . C'est ceci que nous appelons $\Phi(\pi)$ (voir Figure 1.1, à droite).

Observez que chaque point de la permutation donne lieu à un arc de $\Phi(\pi)$: nous dirons alors que le point *correspond* à l'arc et vice-versa.

Théorème 1.1. *L'application Φ est une bijection entre les permutations de Baxter et les orientations bipolaires planes qui agit sur les paramètres standards de la manière suivante :*

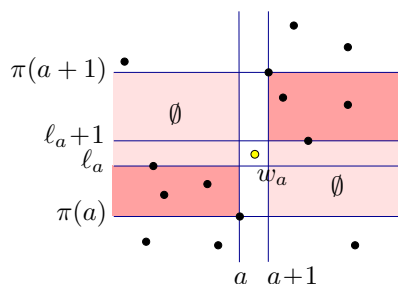


FIGURE 1.5 – L’insertion d’un sommet blanc à la place d’une montée.

$$\begin{array}{l|l}
 \text{taille} \leftrightarrow \# \text{ arcs,} & \# \text{ montées} \leftrightarrow \# \text{ sommets non-polaires,} \\
 \# \text{ lr-maxima} \leftrightarrow \text{degré sortant gauche,} & \# \text{ rl-minima} \leftrightarrow \text{degré sortant droit,} \\
 \# \text{ rl-maxima} \leftrightarrow \text{degré du puits,} & \# \text{ lr-minima} \leftrightarrow \text{degré de la source.}
 \end{array}$$

Par la formule d’Euler, le nombre de faces internes de $\Phi(\pi)$ est égal au nombre de descentes de π . Le Théorème 1.1 montre pourquoi les nombres (1.1) apparaissent aussi bien dans l’énumération des permutations de Baxter [Mal79] que dans l’énumération des orientations bipolaires planes [Bax01].

1.3.2 La bijection inverse

Soit O une orientation bipolaire plane dont les sommets sont coloriés en blanc.

Soit O' la carte planaire orienté bicolore obtenue à partir de O en subdivisant chaque arc en 2 arcs avec un sommet noir au milieu (Figure 1.6). Il faut se souvenir comment les arcs entrants et sortants sont répartis autour des sommets de O (Figure 1.3, à droite).

Soit T_x la sous-carte de O' obtenue en gardant pour chaque sommet blanc le premier arc entrant dans le sens horaire (pour le puits, nous gardons seulement l’arc entrant du bord droit). Chaque sommet de T_x , à l’exception de la source de O , a degré entrant 1 et peut atteindre la source. Donc T_x est un arbre plan enraciné sur la source de O .

De manière similaire T_y est obtenu à partir de O' en ne gardant que le dernier arc entrant de chaque sommet blanc (pour le puits, nous ne gardons que l’arc entrant appartenant au bord gauche). T_y est aussi un arbre plan enraciné sur la source de O .

Maintenant étiquetons les sommets noirs de T_x dans l’ordre préfixe en tournant autour de l’arbre T_x dans le sens horaire. De manière similaire les sommets noirs de T_y sont étiquetés dans l’ordre préfixe mais cette fois-ci en tournant autour de l’arbre dans le sens anti-horaire. Pour chaque sommet noir v de O' , on crée alors un point de coordonnées $(x(v), y(v))$, où $x(v)$ et $y(v)$ sont les étiquettes de v dans les arbres T_x et T_y respectivement. Soit $\Psi(O)$ la collection des points ainsi obtenus (un par arête de O).

Théorème 1.2. *Pour chaque orientation bipolaire plane O , l’ensemble de points $\Psi(O)$ est le diagramme de la permutation de Baxter de $\Phi^{-1}(O)$.*

On peut vérifier ce résultat sur les exemples des Figures 1.1 et 1.6. La construction Ψ est très proche de l’algorithme de dessin de graphes présenté dans [DBTT92].

1.3.3 Symétries de la bijection

Nous allons maintenant décrire comment les symétries du carré, qui agissent naturellement sur les permutations de Baxter, sont transformées via notre bijection.

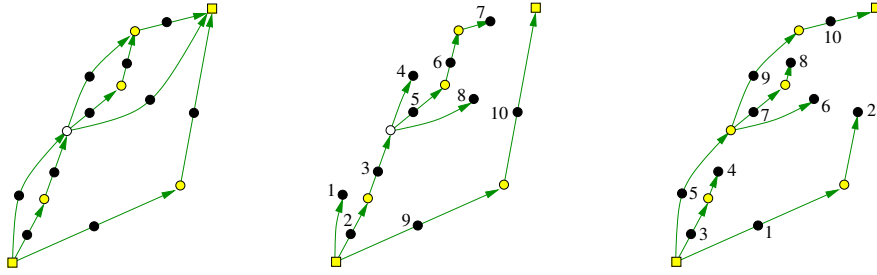


FIGURE 1.6 – Une orientation bipolaire O , et les arbres T_x et T_y utilisés pour calculer les coordonnées des points de $\Psi(O)$.

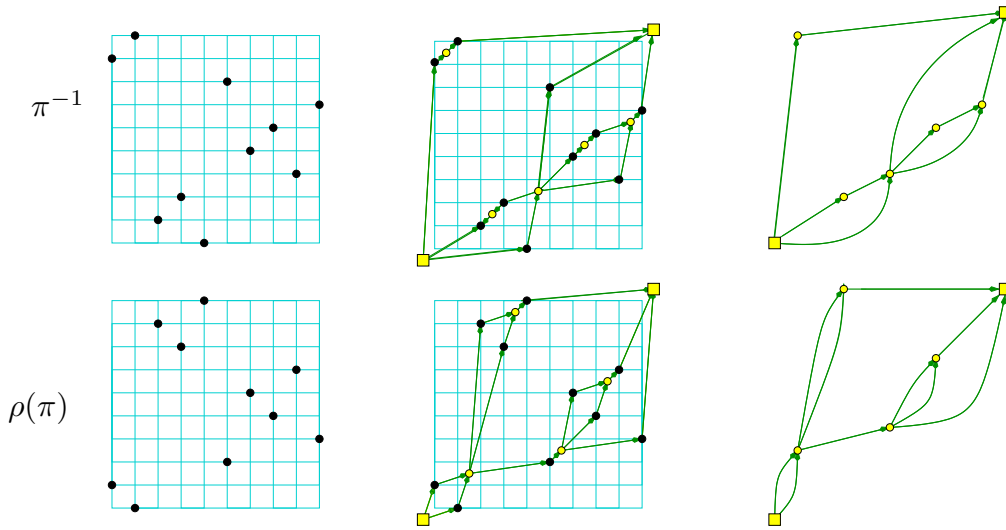


FIGURE 1.7 – Le diagramme de Hasse et l'orientation bipolaire associée à π^{-1} et $\rho(\pi)$, où $\pi = 53497810612$ est la permutation de Baxter de la Figure 1.1.

Proposition 1.2. Soit π une permutation de Baxter et $O = \Phi(\pi)$. Alors

$$\Phi(\pi^{-1}) = \text{mir}(O) \quad \text{et} \quad \Phi(\text{rev}(\pi)) = \text{mir}(O^*).$$

En combinant ces 2 propriétés, cela implique

$$O^* = \Phi(\rho(\pi)),$$

où ρ est la rotation de 90 degrés dans les sens horaire.

De plus, si le point $p = (i, \pi(i))$ de π correspond (via la bijection Φ) à l'arc e de O , alors le point $(\pi(i), i)$ de π^{-1} correspond à l'arc $\text{mir}(e)$ dans $\text{mir}(O)$ et le point $\rho(p)$ dans $\rho(\pi)$ correspond à l'arc dual de e dans O^* .

Les propriétés de symétrie faisant intervenir l'inverse et la rotation sont illustrées dans la Figure 1.7. La première se prouve facilement à partir des définitions de ϕ et Φ . En fait, il est clair en voyant la Figure 1.5 que π a une montée a si et seulement si π^{-1} a une montée ℓ_a et que les points blancs du graphe plongé $\phi(\pi^{-1})$ sont les images miroir des points blancs de $\phi(\pi)$ par rapport à la première diagonale. Bien sûr cela est également vrai pour les points noirs. Donc le diagramme de Hasse de $\phi(\pi^{-1})$ est obtenu en flippant le diagramme de $\phi(\pi)$ par rapport à la première diagonale. La première propriété s'en suit tout comme la correspondance entre le point $(\pi(i), i)$ et l'arc $\text{mir}(e)$.

1.4 Spécialisations

Comme rappelé dans la section 1.2, une carte bipolaire M admet une orientation bipolaire si et seulement si \hat{M} est non-séparable.

Prenons une permutation de Baxter π et l'orientation bipolaire $\Phi(\pi)$, avec comme pôles s et t . Soit M la carte bipolaire sous-jacente et définissons $\hat{\Phi}(\pi)$ la carte enracinée non-séparable \hat{M} . Nous appelons toujours s et t la source et le puits de la carte enracinée. Il est facile de voir que différentes permutations de Baxter peuvent donner la même carte. Cependant la proposition suivante est vraie :

Proposition 1.3. *La restriction de $\hat{\Phi}$ aux permutations de Baxter évitant le motif 2413 (c'est-à-dire les permutations évitant les motifs 2413 et 41352) est une bijection entre ces permutations et les cartes non-séparables planaires enracinées. De plus $\hat{\Phi}$ agit sur les paramètres standards de la manière suivante :*

- si π est de taille n , avec m montées, i lr-maxima, j rl-maxima, k lr-minima et ℓ rl-minima,
- alors $\hat{\Phi}(\pi)$ a $n + 1$ arcs, m sommets non-polaires, un puits de degré $j + 1$, une source de degré $k + 1$ et la face à droite (resp. gauche) de l'arête racine est de degré $i + 1$ (resp. $\ell + 1$).

Notons que le fait que les permutations qui évitent les motifs 2413 et 41352 sont énumérées comme les cartes planaires non-séparables était déjà prouvé dans [DGW96]. Ce résultat était obtenu en exhibant deux arbres de génération isomorphes, un pour chaque classe d'objets. Cet isomorphisme peut être utilisé pour décrire récursivement une bijection entre les permutations et les cartes. Il apparaît que via une simple symétrie, notre bijection est équivalente à celle implicite de [DGW96].

Observons que si π est une permutation de Baxter qui évite le motif 2413, alors π^{-1} est une permutation de Baxter qui évite le motif 3142. Comme $\Phi(\pi^{-1}) = \text{mir}(\Phi(\pi))$, la restriction de $\hat{\Phi}$ aux permutations de Baxter qui évitent 3142 est aussi une bijection avec les cartes planaires non-séparables. Nous décrivons maintenant ce qui se passe quand on restreint $\hat{\Phi}$ aux permutations qui évitent 2413 et 3142 (ces permutations sont toujours de Baxter car elles évitent forcément les motifs 25314 et 41352).

Proposition 1.4. *La restriction de $\hat{\Phi}$ aux permutations qui évitent 2413 et 3142 est une bijection entre ces permutations et les cartes série-parallèles enracinées. De plus les paramètres standards de ces objets sont transformés comme indiqué dans la Proposition 1.3.*

On dit qu'une carte enracinée M est *série-parallèle* si elle ne contient pas K_4 comme mineur. Cette terminologie peut induire en erreur : la carte qui peut être construite récursivement en utilisant les constructions classiques série et parallèle (Figure 1.10) n'est pas la carte M elle-même, mais une carte bipolaire \tilde{M} .

Il est facile d'énumérer les cartes série-parallèles en utilisant leur description récursive (on se trouve ici dans le cadre simple des *structures décomposables* [FS09]). Le nombre de permutations de taille n qui évitent les motifs 2413 et 3142 est le n -ème nombre de Schröder :

$$\sum_{k=0}^n \frac{(n+k)!}{k!(k+1)!(n-k)!}$$

Ce résultat a été prouvé ici [Wes95]. Une preuve bijective de ce résultat est également proposée [Gir93].

1.4.1 Permutations de Baxter évitant le motif 2413 et les cartes enracinées non-séparables

Le but de cette sous-section est de prouver la Proposition 1.3 : si on considère la restriction de Φ aux permutations de Baxter qui évitent 2413, qu'on ajoute un arc racine de la source au puits et qu'on oublie l'orientation des arcs (sauf celle de l'arc racine), on obtient une bijection avec les cartes planaires non-séparables enracinées. De plus cette bijection envoie les paramètres standards des permutations sur des paramètres standards des cartes. Notre objectif premier sera de décrire les orientations correspondantes,

via Φ aux permutations de Baxter qui évitent 2413. Tout d'abord rappelons-nous que chaque face d'une orientation bipolaire plane possède un sommet gauche et un sommet droit (Figure 1.3). Les définitions suivantes sont illustrées dans la Figure 1.8(a).

Définition 1.1. *Étant donnée une orientation bipolaire plane O , une pièce orientée à droite (ROP, pour right-oriented-piece) est le quadruplet (v_1, v_2, f_1, f_2) formé de 2 sommets v_1, v_2 et de 2 faces internes f_1, f_2 de O tel que :*

- v_1 est la source de f_1 et un sommet gauche de f_2 ,
- v_2 est le puits de f_2 et un sommet droit de f_1 .

Une pièce orientée à gauche (LOP pour left-oriented-piece) est définie de manière similaire en inversant le rôle de "gauche" et "droite" dans la définition de ROP. Clairement une ROP dans O devient une LOP dans $\text{mir}(O)$ et vice-versa.

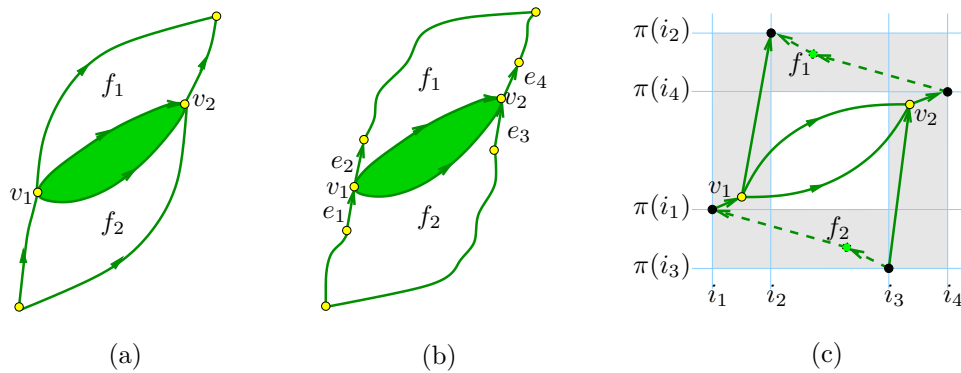


FIGURE 1.8 – (a) Une pièce orientée à droite (ROP) d'une orientation bipolaire plane. (b) Les 4 arcs distingués d'une ROP. (c) Le motif minimal 2413 dans une permutation de Baxter se traduisant par une ROP dans l'orientation bipolaire plane associée. Les arêtes en pointillés viennent de $\phi(\rho(\pi))$.

Proposition 1.5 ([OdM94]). *Chaque carte planaire bipolaire admet une unique orientation bipolaire sans ROP et une unique orientation bipolaire sans LOP.*

L'ensemble des orientations bipolaires d'une carte bipolaire fixée peut être équipée d'une structure de treillis distributif. Le minimum (resp. maximum) de ce treillis est l'unique orientation sans ROP (resp. LOP) [OdM94].

Nous pouvons maintenant caractériser l'image par Φ des permutations de Baxter évitant le motif 2413.

Proposition 1.6. *Une permutation de Baxter π contient le motif 2413 si et seulement si l'orientation bipolaire $O = \Phi(\pi)$ contient une ROP. De manière analogue, π contient le motif 3142 si et seulement si O contient une LOP.*

Les permutations $\pi = 25314$ et $\pi = 41352$ fournissent des exemples simples. Les orientations correspondantes (avec l'ajout de l'arc racine) sont celles de la Figure 1.9.



FIGURE 1.9 – Les 2 orientations bipolaires de K_4 . La première contient une ROP et la seconde une LOP.

Preuve de la Proposition 1.3. La restriction de Φ aux permutations de Baxter qui évitent 2413 est une bijection entre ces permutations et les orientations bipolaires planes sans ROP. Par la proposition 1.5, les orientations sans ROP sont en bijection avec les cartes planaires non-séparables (cette bijection consiste en l'ajout d'un arc racine entre la source et le puits et l'oubli de l'orientation des arcs à l'exception de l'arc racine). Cette bijection incrémente le nombre d'arêtes, le degré de la source et du puits de 1. De plus elle transforme les degrés extérieurs droite et gauche de l'orientation en les degrés des faces à gauche et à droite de l'arc racine moins 1. La proposition 1.3 en découle. \square

1.4.2 Permutations évitant les motifs 2413 et 3142 et les cartes série-parallèles

Nous allons maintenant prouver la Proposition 1.4. Une carte planaire non-séparable M est *série-parallèle* si elle ne contient pas K_4 comme mineur. Soit \check{M} la carte bipolaire correspondante. Nous dirons que \check{M} est elle-même série-parallèle. En adaptant la preuve sur les graphes de [BGKN07], on peut facilement voir qu'une carte bipolaire \check{M} est série-parallèle si et seulement si elle peut être construite récursivement en partant d'une carte réduite à simple arc et en appliquant une séquence de compositions séries et parallèles :

- la *composition série* de 2 cartes bipolaires série-parallèles \check{M}_1 et \check{M}_2 est obtenue en identifiant le puits de \check{M}_1 avec la source de \check{M}_2 (Figure 1.10 (b)),
- la *composition parallèle* de 2 cartes bipolaires série-parallèles \check{M}_1 et \check{M}_2 est obtenue en plaçant \check{M}_2 à droite de \check{M}_1 puis en identifiant les sources de \check{M}_1 et de \check{M}_2 et en identifiant également leur puits (Figure 1.10 (c)).

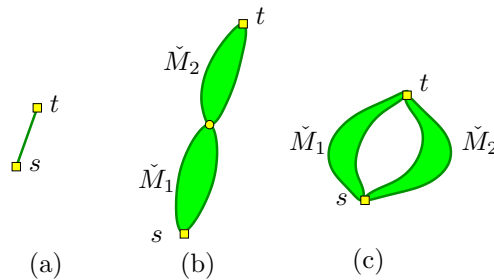


FIGURE 1.10 – Les opérations permettant de construire des cartes bipolaires série-parallèles : (a) prendre un arc, (b) la composition série, (c) la composition parallèle.

Par la proposition 1.6, une permutation de Baxter évite les 2 motifs 2413 et 3142 si et seulement si l'orientation bipolaire plane correspondante ne possède ni ROP ni LOP. Nous avons vu dans la Proposition 1.5 qu'une carte bipolaire \check{M} admet une unique orientation sans ROP et une unique orientation sans LOP, qui sont respectivement les éléments maximal et minimal du treillis des orientations de \check{M} . Donc \check{M} admet une orientation sans ROP et sans LOP si et seulement si \check{M} admet une unique orientation bipolaire. Donc pour montrer la Proposition 1.4, il suffit de montrer le lemme suivant.

Lemme 1.1. *Une carte bipolaire \check{M} admet une unique orientation bipolaire si et seulement si elle est série-parallèle.*

Preuve A l'aide de la construction récursive des cartes bipolaires série-parallèles il est facile de vérifier que ces cartes admettent une unique orientation bipolaire (voir aussi [dFOdMR95, Remark 6.2]).

Réciproquement, supposons que \check{M} n'est pas série-parallèle. Cela signifie que la carte enracinée sous-jacente M contient K_4 comme mineur. On peut observer que K_4 admet exactement 2 orientations bipolaires comme le montre la Figure 1.9. D'après le *Lemme d'extension* de [dFOdMR95], ces 2 orientations bipolaires de K_4 peuvent être étendues à 2 orientations bipolaires distinctes de M et donc de \check{M} . \square

Ceci conclut la preuve de la Proposition 1.4.

1.5 Digression sur les spécialisations

Avant de conclure ce chapitre, je souhaiterais revenir sur le fait que les spécialisations de la bijection Φ en dévoilent sa richesse et qu'avec Olivier Bernardi nous avons appliqué cette même démarche pour une autre bijection.

Dans le cas décrit dans ce chapitre, nous sommes en présence d'une famille de cartes (les cartes bipolaires planaires non-séparables) munies d'une orientation contrainte (les orientations bipolaires). L'ensemble des orientations d'une carte bipolaire fixée forme un treillis distributif, ainsi chaque carte possède un unique élément maximal (sans ROP) et un unique élément minimal (sans LOP). De plus il existe une sous famille de cartes admettant une unique orientation bipolaire (donc sans ROP ni LOP). La bijection Φ se comporte très bien vis-à-vis de cette structure de treillis distributif des orientations bipolaires. Cette bijection se spécialise 2 fois : une fois en considérant les orientations sans LOP (par exemple) et une deuxième fois en considérant les orientations sans LOP ni ROP.

Comme nous l'avons montré dans [BB09], les 3-orientations définies sur les triangulations du plan se trouvent dans le même cas de figure. Une 3-orientation d'une triangulation est une orientation des arêtes internes telle que chaque sommet interne soit de degré sortant 3. L'ensemble des 3-orientations d'une triangulation du plan a une structure de treillis distributif [OdM94, Pro93]. L'élément maximal (resp. minimal) de ce treillis est la 3-orientation ne contenant pas de circuit tournant dans le sens horaire (resp. anti-horaire). Les *bois de Schnyder* (que nous définirons formellement Section 2.2) qui sont en bijection avec les 3-orientations : il s'agit d'une coloration canonique des arcs de la 3-orientation de sorte que : chaque sommet interne ait un arc sortant de chaque couleur (0,1,2) et que les arcs entrants situés entre l'arc situés entre les arcs sortant colorié i et $i + 1$ dans le sens anti-horaire sont entrants et coloriés $i - 1$ (nous supposons qu'il y a un ordre cyclique sur les couleurs 0, 1, 2 de sorte que $i + 1$ et $i - 1$ soient toujours définis). La Figure 1.11 montre un exemple de bois de Schnyder.

L'application présentée dans [Bon05] (et décrite de manière simplifiée dans [BB09]) envoie les 3-orientations des triangulations ayant n sommets internes sur des couples de chemins de Dyck de taille n .

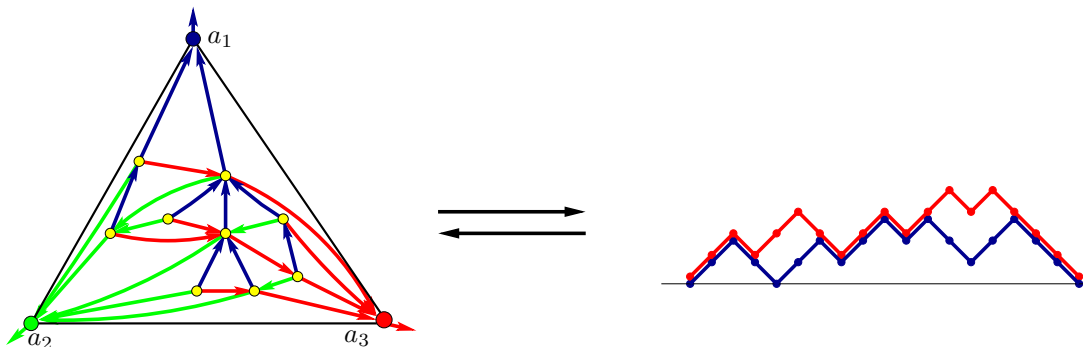


FIGURE 1.11 – Illustration de la bijection entre les bois de Schnyder et intervalles de Stanley. Le chemin du bas (bleu) code l'arbre bleu et la longueur des descentes du chemin du haut (rouge) code degré entrant rouge de chaque sommet. A partir de ces 2 information on peut retrouver de manière unique les arbres bleu et rouge entremêlés. Puis il y a une unique manière d'y ajouter l'arbre vert.

Un *chemin de Dyck* P de taille n est un chemin composé de n pas nord-est (noté N) et de n pas sud-est (noté S) partant du point $(0, 0)$ arrivant au point $(2n, 0)$ et restant dans le quart de plan positif. Étant donné une relation d'ordre \leq sur les chemins de Dyck de taille n , un *intervalle* est une paire de chemins P, Q tels que $P \leq Q$. La famille des objets de Catalan (et donc l'ensemble des chemins de Dyck qui fait partie de cette grande famille) est équipé de trois relations d'ordre remarquables formant des treillis : le *treillis de Stanley*, le *treillis de Tamari* ainsi que le *treillis de Kreweras*. Le treillis de Stanley

est une extension du treillis de Tamari, lui-même est une extension du Treillis de Kreweras. Ces 3 treillis peuvent être définis simplement par leurs relations de couverture :

- Q couvre P dans le treillis de Stanley, si Q peut-être obtenu à partir de P en remplaçant un sous-mot SN par un NS .
- Q couvre P dans le treillis de Tamari, si Q peut-être obtenu à partir de P en déplaçant un pas S après le sous-mot *premier* (s’il existe) qui le suit directement. Rappelons qu’un sous-mot est dit *premier* s’il contient autant de pas N que de pas S et que tout préfixe non trivial de ce sous-mot contient strictement plus de pas N que de pas S .
- Q couvre P dans le treillis de Kreweras, si Q peut-être obtenu à partir de P en déplaçant un pas S après un sous-mot de Dyck (s’il existe) qui le suit directement. Rappelons qu’un sous-mot est dit de Dyck s’il contient autant de pas N que de pas S et que tout préfixe de ce sous-mot contient au moins autant de pas N que de pas S .

La bijection que nous avons étudiée se comporte remarquablement bien vis-à-vis de l’inclusion de ces trois treillis. En effet :

- les 3-orientations en bijection avec les intervalles de Stanley
- les 3-orientations minimales en bijection avec les intervalles de Tamari.
- les 3-orientations minimales et maximales sont en bijection avec les intervalles de Kreweras.

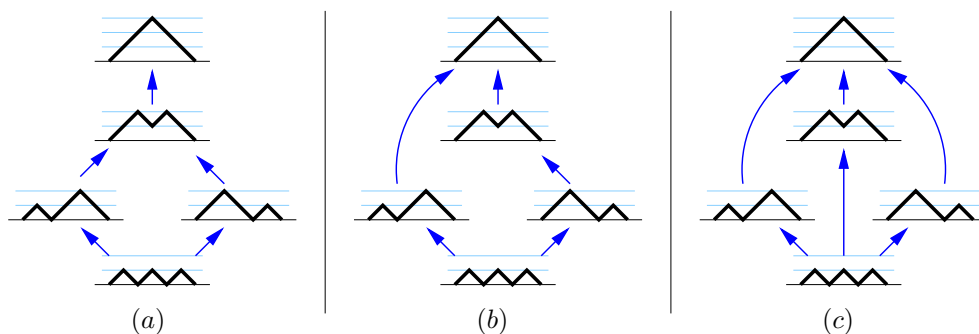


FIGURE 1.12 – Diagramme de Hasse des treillis de Catalan sur l’ensemble des chemins de Dyck de taille 3. (a) Treillis de Stanley. (b) Treillis de Tamari. (c) Treillis de Kreweras.

Les 3-orientations minimales sont trivialement en bijection avec les cartes sous-jacentes, à savoir les triangulations du plan. Ceci permet de montrer bijectivement un résultat d’énumération sur les intervalles de Tamari découvert par Chapoton [Cha06]. Nous montrons également que les triangulations admettant une unique 3-orientation sont exactement les triangulations *3-dégénérées* (ou *stack triangulations*) c’est à dire qui peuvent être récursivement épluchées par suppression des sommets de degré 3. Ces triangulations sont connues pour être en bijection avec les arbres ternaires. On retrouve ainsi des résultats connus, mais dans un cadre unifié.

1.6 Conclusion et perspectives

Au regard de ces 2 exemples de spécialisation, on se rend compte que l’enrichissement des cartes par une structure de treillis distributif (ici des orientations contraintes) apporte un éclairage sur les bijections existantes sur les cartes (par exemple entre les cartes planaires non-séparables et les permutations à motifs exclus) ou permet même de découvrir de nouvelles bijections (par exemple entre les triangulations du plan et les intervalles de Tamari).

Parmi les treillis distributifs que l’on peut définir sur les cartes, citons par exemple les modèles de tas de sable [GLM⁺04] qui regroupent entre autre les “jeux par retournement de puits” (*edge firing game* en anglais) [LM02], les arbres couvrants, les orientations Eulériennes (*i.e.* pour chaque sommet son degré entrant est égal à son degré sortant). Le cas de cartes équipées d’un arbre couvrant a déjà été étudié

avec succès par Bernardi [Ber07]. On peut espérer des résultats similaires pour d'autres familles de cartes supportant d'autres treillis distributifs.

Prenons un premier exemple concret pour illustrer cela. Récemment, Fusy [Fus10] a étendu la bijection présentée dans ce chapitre pour montrer que les involutions de Baxter sans point fixe de taille $2n$ sont en bijection avec les orientations planaires à n arêtes ayant une unique source et un ou plusieurs puits, tous situés sur la face externe. Ceci montrant bijectivement que ces permutations sont énumérées par :

$$\frac{3 \cdot 2^{n-1}}{(n+1)(n+2)} \binom{2n}{n}.$$

De manière surprenante, ces nombres énumèrent également deux autres familles de cartes et une autre famille de permutations. Il s'agit des cartes Eulériennes à n arêtes, des cartes bicubiques (i.e. cartes biparties dont tous les sommets sont de degré 3) à $3n$ arêtes et des permutations *indécomposables* (qui ne peut pas être découpées en deux parties telles que toute la partie avant la coupe est supérieure à la partie après la coupe) évitant le motif 1342. Une preuve bijective de l'énumération de ces 2 familles de cartes a été proposée par Cori, Jacquard et Schaeffer [CJSJ97] en utilisant les *arbres de description*. Bóna [B97] a ensuite proposé une bijection entre les cartes bicubiques et les permutations indécomposables évitant le motif 1342. Cette bijection se base sur les arbres de description. Dans ce contexte, l'étude des orientations Eulériennes planes pourrait éclairer ces résultats et idéalement permettrait de faire apparaître une bijection entre les orientations Eulériennes et une famille de permutations incluant soit les permutations indécomposables évitant le motif 1342, soit les involutions de Baxter sans point fixe. Cette bijection plus générale pourrait alors se spécialiser pour retrouver le résultat de Bóna ou faire un pont avec les involutions de Baxter sans point fixe.

De manière plus prospective, les cartes de genre supérieur sont un terrain de jeux très prometteur. Les travaux de Chapuy [Cha11] montrent que ces cartes semblent nettement plus complexes à étudier que les cartes planaires. On peut espérer que certaines familles de cartes de genre supérieur (par exemple les cartes unicellulaires ou les triangulations) enrichies de la bonne structure de treillis distributif soient plus abordables. Deux généralisations des bois de Schnyder ont été proposées : une première pour les triangulations toriques [AFL09] et une seconde pour les triangulations de genre supérieur [GL12]. L'étude de la structure de ces objets pourrait éclairer l'étude des cartes de genre supérieur.

Chapitre 2

Dessin de graphes : Dessin convexe de cartes planaires 3-connexes

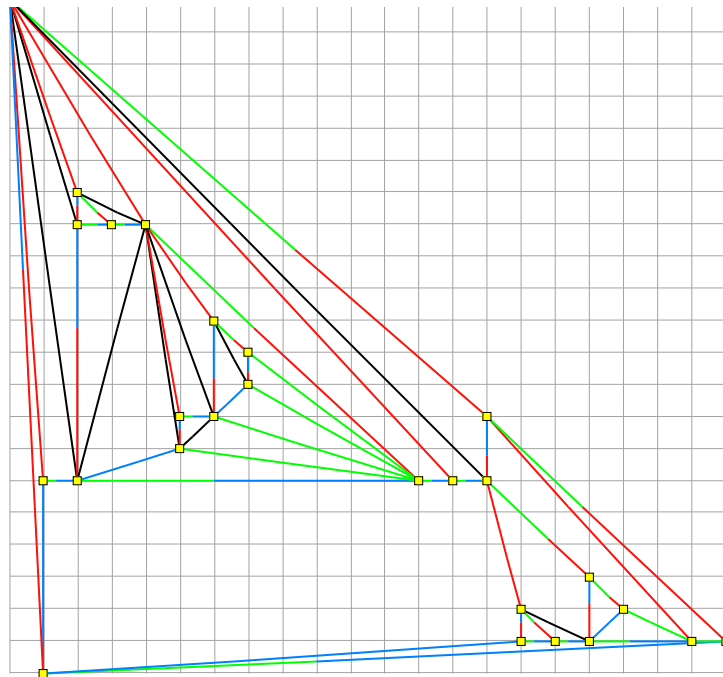


FIGURE 2.1 – Exemple de dessin de graphe 3-connexe obtenu pour à l'aide de notre algorithme. Ce graphe possède 26 sommets, 57 arêtes et 33 faces. Il est dessiné sur une grille de taille 21×21 .

Dans ce chapitre nous nous intéressons au dessin de graphes. Il existe une très grande diversité dans les modèles de représentation de graphes. Par exemple dans les *dessins de visibilité*, les sommets sont représentés par des segments horizontaux et les arêtes par des segments verticaux, dans les *dessins de 2-visibilité* les sommets sont représentés par des rectangles et les arêtes par des segments horizontaux ou verticaux, dans le modèle *dessin par contact* les sommets sont représentés par des formes géométriques (des cercles, des triangles, ...) et les arêtes sont symbolisées par des contacts entre ces formes. Ici nous nous intéressons au modèle de représentation le plus simple : *dessin en lignes droites* ou *dessin de Fary* (*straight-line drawing* en anglais). Dans ce modèle les sommets sont représentés par des points à coordonnées entières et les arêtes par des segments de droite joignant ces points. Dessiner un graphe revient donc à calculer les coordonnées de chaque sommet. Bien évidemment dans les cas de graphes planaires, ces segments ne peuvent s'intersecter qu'à leurs extrémités.

Ce chapitre présente un algorithme permettant d’obtenir pour des graphes planaires 3-connexes des dessins compact de Fary où les faces sont convexes. Ce travail a été réalisé en collaboration avec Stefan Felsner (TU Berlin) et Mohamed Mosbah [BFM07]. L’étude en moyenne de la taille des dessins obtenus sur les triangulations est une conséquence des travaux que nous avons menés avec Cyril Gavoille, Nicolas Hanusse, Dominique Poulalhon (LIX - Palaiseau) et Gilles Schaeffer (LIX - Palaiseau) [BGH⁺06].

2.1 Introduction

Depuis 1948 [Far48] on sait que tout graphe planaire admet un dessin en lignes droites. Rosenstiehl et Tarjan [RT86] ont soulevé la question de savoir s’il était possible d’obtenir de tels dessins sur des grilles de taille polynomiale. La première réponse positive a été donnée par de Fraysseix, Pach et Pollack [FPP88], [FPP90]. L’algorithme qu’ils ont proposé produit des dessins de taille $(2n-4) \times (n-2)$ en temps $O(n \log n)$. De manière indépendante, Schnyder [Sch89] a proposé un algorithme (algorithme par *comptage de faces*) linéaire produisant des dessins de taille $(2n-5) \times (2n-5)$. Par la suite Schnyder [Sch90] a proposé un algorithme produisant des dessins de taille $(n-2) \times (n-2)$. Récemment, Zhang et He [ZH03] en utilisant les bois de Schnyder minimaux (objet que nous n’allons pas tarder à définir), ont montré que toute triangulation peut être dessinée sur des grilles de taille $(n-1-\Delta^{\oplus}) \times (n-1-\Delta^{\oplus})$, où Δ^{\oplus} est un paramètre positif du bois de Schnyder minimal de la triangulation.

Comme l’a montré Tutte [Tut60], les graphes planaires 3-connexes (*i.e.* graphes qui restent connexes lorsqu’on supprime 2 sommets) admettent un *dessin convexe*, c’est à dire tel que les arêtes bordant une face forment un polygone convexe. Kant [Kan96] a étendu l’approche de de Fraysseix et al [FPP88] pour construire des dessins convexes de taille $(2n-4) \times (n-2)$. Plus tard, Chrobak et Kant [CK97] ont réussi à réduire la taille des dessins à $(n-2) \times (n-2)$. De manière indépendante Di Battista et al. [BTV99] et Felsner [Fel01] ont proposé une extension aux graphes 3-connexes de l’algorithme par comptage de faces de Schnyder [Sch89] produisant des dessins de taille $(f-1) \times (f-1)$ où f est le nombre de faces du graphe.

Il est important de distinguer les dessins convexes des dessins strictement convexes. Dans un dessin strictement convexe, chaque angle intérieur doit être strictement inférieur à π et chaque angle extérieur doit être strictement supérieur à π . Les dessins strictement convexes ont été étudiés par Rote [Rot05]. Il a montré qu’il était possible de dessiner des graphes planaires 3-connexes de manière strictement convexe sur des grilles de taille $O(n^{7/3}) \times O(n^{7/3})$.

Dans ce chapitre, nous nous intéressons aux dessins convexes. Nous proposons une extension de l’algorithme par comptage de faces qui produit des dessins convexes de taille $(n-1-\Delta) \times (n-1-\Delta)$, où $\Delta \geq \max\{0, n-f\}$ est un paramètre du bois de Schnyder minimal du graphe 3-connexe.

La suite de ce chapitre est organisée de la manière suivante. Dans la Section 2.2, nous introduisons la notion de bois de Schnyder et nous montrons comment cette structure combinatoire est utilisée par l’algorithme de dessin *par comptage de faces*. Nous présentons également la structure de treillis distributif des bois de Schnyder d’un graphe et nous présentons une nouvelle opération appelée *fusion* qui transforme à la fois le bois de Schnyder et le graphe sous-jacent.

La Section 2.3 présente notre algorithme de dessin convexe qui utilise les fusions pour réduire la taille du dessin. La Section 2.4 présente quelques idées qui permettent de réduire encore un peu la taille des dessins. Nous présentons également quelques résultats expérimentaux qui montrent que les dessins que nous obtenons sont en moyenne significativement plus compacts que ceux des algorithmes précédents.

2.2 Bois de Schnyder et algorithme par comptage de faces

2.2.1 Définitions

Schnyder a défini une coloration et une orientation spéciale des arêtes internes d’une triangulation. Dans [Sch89] et [Sch90] il a utilisé ces objets pour caractériser les graphes planaires en termes de dimension

d'ordre puis pour dessiner ces graphes sur des petites grilles. Nous décrivons ici une généralisation de ces bois de Schnyder aux cartes planaires 3-connexes. Cette généralisation a été présentée dans [BTV99] et [Fel01]. Nous utilisons ici les notations introduites dans [Fel04a].

Les cartes que nous allons considérer dans ce chapitre sont simples (*i.e.* sans boucle ni arête multiple).

Une *suspension* M^σ d'une carte M est obtenue en distinguant 3 sommets a_1, a_2 et a_3 , ces sommets de la face externe appelés *sommets de suspension*; ces 3 sommets apparaissent dans cet ordre, lorsqu'on fait le tour de la face externe dans le sens horaire. De plus nous attachons à chaque sommet de suspension une demi-arête dans la face externe.

Dans la suite les indices 1,2 et 3 seront associés respectivement avec les couleurs bleu, rouge et vert.

Définition 2.1. Soit M^σ une carte suspendue. Un bois de Schnyder (Schnyder wood¹ en anglais) est une orientation et une coloration des arêtes de M^σ avec 3 couleurs 1, 2 et 3 qui vérifie les règles suivantes :

- (W1) Chaque arête e est simplement orientée ou orientée dans les 2 directions opposées. Si l'arête est simplement orientée alors elle est coloriée d'une seule couleur, sinon (elle est doublement orientée) elle est coloriée avec 2 couleurs.
- (W2) Chaque demi arête de a_i est sortante de a_i est coloriée i .
- (W3) Chaque sommet v a un degré sortant de 1 dans chaque couleur. De plus les arêtes e_1, e_2, e_3 sortant de v sont coloriés 1, 2 et 3 dans le sens horaire. Chaque arête entrante de v coloriée i se trouve entre l'arête e_{i-1} et e_{i+1} dans le sens horaire. Voir Figure 2.2.
- (W4) Il n'y a pas de face interne possédant un circuit orienté d'une seule couleur.

Remarque : lorsque la carte considérée est une triangulation, toutes les arêtes internes sont simplement orientées et les 3 arêtes externes sont doublement orientées. A l'exception des 3 arêtes externes qui sont orientées et coloriées de manière canonique, la définition des bois de Schnyder ci-dessus coïncide avec la définition proposée par Schnyder [Sch89] pour les triangulations.

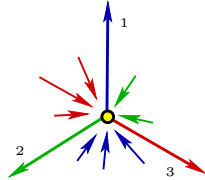


FIGURE 2.2 – Coloration et orientation des arêtes autour d'un sommet.

Une carte est *internement 3-connexe* si l'ajout du sommet v_∞ connecté à tous les sommets de la face externe donne une carte planaire 3-connexe. Une suspension est *internement 3-connexe* si l'ajout d'un sommet v_∞ connectés aux sommets de suspension par les 3 demi-arêtes donne une carte planaire 3-connexe.

Fait 1. (cf. Théorème 5.1 de [Mil02]) Il existe un bois de Schnyder pour M^σ , si et seulement si M^σ est une suspension internement 3-connexe.

Étant donné un bois de Schnyder d'une suspension M^σ , on note T_i l'ensemble des arêtes coloriées i .

Fait 2. T_i est un arbre couvrant de M^σ . De plus si on enracine cet arbre en a_i , les arêtes sont orientées des enfants vers leur parent.

On note $P_i(v)$ le chemin de v à a_i dans T_i .

1. Dans la littérature on retrouve ces objets sous le nom de *realizer*, *Schnyder trees* ou *Schnyder labeling*. Le terme *Schnyder wood* introduit par Felsner [Fel04b] semble le plus approprié car il permet de mettre en évidence qu'il s'agit d'un ensemble d'arbres (un *bois*) partageant le même ensemble de sommets, contrairement à une forêt, où les différents arbres composant la forêt ont des ensembles de sommets disjoints.

2.2.2 Dessin convexe par comptage de faces

Cette approche a été décrite dans [BTV99] et [Fel01]. Elle peut être vue comme une extension aux cartes intérieurement 3-connexe de l'algorithme initial de Schnyder [Sch89] sur les triangulations.

A partir de la condition (W3) on a que pour $i \neq j$ les chemins $P_i(v)$ et $P_j(v)$ n'ont qu'un seul sommet en commun, le sommet v . Donc les chemins $P_1(v), P_2(v), P_3(v)$ divise M en 3 régions $R_1(v), R_2(v)$ et $R_3(v)$, où $R_i(v)$ est la région bornée par les chemins $P_{i-1}(v), P_{i+1}(v)$ et $P_{i+1}(a_{i-1})$ ², voir Figure 2.3.

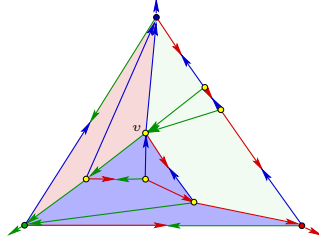


FIGURE 2.3 – Un bois de Schnyder et les régions de v .

- Fait 3.** (a) $R_i(u) \subseteq R_i(v)$ si et seulement si $u \in R_i(v)$.
 (b) $R_i(u) = R_i(v)$ si et seulement si il y a un chemin composé d'arêtes bicolores $i - 1, i + 1$ connectant u à v .
 (c) Pour tout u, v il existe i et j tels que $R_i(u) \subset R_i(v)$ et $R_j(v) \subset R_j(u)$.

Le comptage de faces du sommet v est le vecteur (v_1, v_2, v_3) , où v_i est défini par :

$$v_i = \text{Nombre de faces de } M \text{ dans la région } R_i(v).$$

La propriété d'inclusion des trois régions d'un sommet implique :

- Fait 4.** (a) Si une arête (u, v) est simplement orientée et coloriée i alors : $u_i < v_i, u_{i-1} > v_{i-1}$ et $u_{i+1} > v_{i+1}$.
 (b) Si la demi-arête (u, v) est coloriée $i + 1$ et la demi-arête (v, u) est coloriée $i - 1$ alors $u_i = v_i, u_{i+1} > v_{i+1}$ et $u_{i-1} < v_{i-1}$.

Clairement, pour chaque sommet v , $v_1 + v_2 + v_3 = f - 1$ où f est le nombre de faces de M . Ainsi nous avons une application qui envoie les sommets de M sur le plan $T_f = \{(x_1, x_2, x_3) : x_1 + x_2 + x_3 = f - 1\}$ de \mathbb{R}^3 . En reliant les sommets adjacents par des segments de droite on obtient un dessin $\mu(M)$ de M dans le plan T_f .

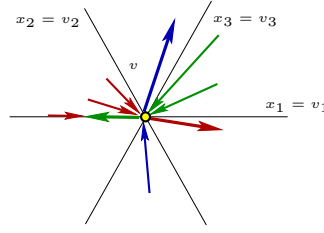
La couleur et l'orientation des arêtes sont codées de manière élégante dans le dessin : Soit v un sommet tel que $\mu(v) = (v_1, v_2, v_3)$. Les 3 lignes $x_1 = v_1, x_2 = v_2$ et $x_3 = v_3$ partitionnent le plan T_f en 6 secteurs de sommet $\mu(v)$. D'après le Fait 4, la couleur et l'orientation des arêtes sont déterminées par le cône qui les contient (voir Figure 2.4). En particulier, les arêtes bicolores sont sur les bords des cônes.

Théorème 2.1. Le dessin $\mu(M)$ est un dessin convexe de la carte M dans le plan T_f . En oubliant la troisième coordonnée, on obtient un dessin convexe de M sur la grille $(f - 1) \times (f - 1)$.

Preuve (schéma)

- $\mu(M)$ est un dessin planaire. Ceci se montre en remarquant que l'algorithme de dessin par comptage de faces produit un plongement *barycentrique faible* défini par Schnyder [Sch90]).

2. Remarque : $P_{i+1}(a_{i-1}) = P_{i-1}(a_{i+1})$

FIGURE 2.4 – Cônes et arêtes autour d'un sommet v dans le plan T_f .

- Les arêtes sortantes autour de chaque sommet (voir Figure 2.4) garantissent que tous les angles internes de $\mu(M)$ sont plus petits que π , et donc que le plongement est convexe.
- La planarité et la convexité sont préservées par projection du plan T_f sur le plan $x_3 = 0$. \square

2.2.3 Treillis des bois de Schnyder

En général une suspension M^σ d'une carte intérieurement 3-connexe admet plusieurs bois de Schnyder. Felsner [Fel04a] a montré que l'ensemble des bois de Schnyder d'une carte intérieurement 3-connexe avait une structure de treillis distributif. Nous allons présenter ici ce résultat.

Pour faire apparaître cette structure il est préférable de considérer conjointement le bois de Schnyder d'une suspension M^σ ainsi que celle de la suspension duale.

La *suspension duale* M^{σ^*} de M^σ est définie de la manière suivante : partant de la suspension M^σ , la carte M^{σ^+} est obtenue en ajoutant un sommet v_∞ relié aux sommets de suspension. Soit b_1, b_2, b_3 les 3 nouvelles faces internes ainsi formées. La suspension M^{σ^*} est obtenue à partir de la carte duale de M^{σ^+} en supprimant le sommet correspondant à la face infinie de M^{σ^+} et en prenant b_1, b_2, b_3 comme sommets de suspension. Pour terminer la construction de M^{σ^*} nous attachons une demi-arête par sommet de suspension dans la face infinie. La Figure 2.5 illustre cette définition.

Remarque : le terme *dual* est impropre car le dual d'une suspension duale n'est pas la suspension primale. En particulier $M^{\sigma^{**}}$ possède 3 sommets de plus que M^σ .

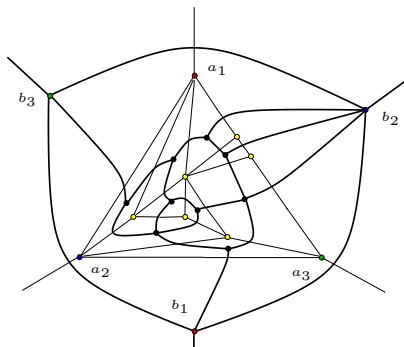


FIGURE 2.5 – La suspension duale de l'exemple de la Figure 2.3.

La *suspension complétée* \widetilde{M}^σ d'une suspension M^σ est une carte obtenue en superposant M^σ avec sa suspension duale M^{σ^*} et en subdivisant chaque arête e de M^σ et sa duale e^* à l'aide d'un sommet commun. Ces nouveaux sommets sont appelés *sommets-arêtes* (représentés par des carrés blancs). Les sommets de M^σ sont dits *primaux* (représentés par des points noirs) et les sommets de M^{σ^*} sont dits *duaux* (représentés par des points blancs). Pour être tout à fait rigoureux, on ajoute des sommets dits

infini (non représentés) à l'extrémité des six demi-arêtes connectées aux sommets de suspension de M^σ et M^{σ^*} . Le plongement de \widetilde{M}^σ se déduit naturellement du plongement de M^σ (et de M^{σ^*}).

Une suspension complétée est planaire. Chaque sommet-arête est de degré 4.

Une 3-orientation d'une suspension complète \widetilde{M}^σ de M^σ est une orientation des arêtes de \widetilde{M}^σ telle que :

- (O1) $\text{outdeg}(v) = 3$ pour les sommets primaux et duaux.
- (O2) $\text{indeg}(v_e) = 3$ for chaque sommet-arête v_e (donc $\text{outdeg}(v_e) = 1$).
- (O3) Les sommets infinis ont degré sortant 0.

Théorème 2.2. *Soit M^σ une suspension d'une carte planaire intérieurement 3-connexe M . Les structures suivantes sont en bijection :*

- (1) les bois de Schnyder M^σ .
- (2) les bois de Schnyder de la suspension duale M^{σ^*} .
- (3) les 3-orientations de la suspension complétée \widetilde{M}^σ .

Ces bijections sont illustrées dans la Figure 2.6. La preuve de ce théorème se trouve dans [Fel04a]. La preuve du Lemme 2.2 montre comment colorier les arêtes de M^σ à partir de la 3-orientation de \widetilde{M}^σ .

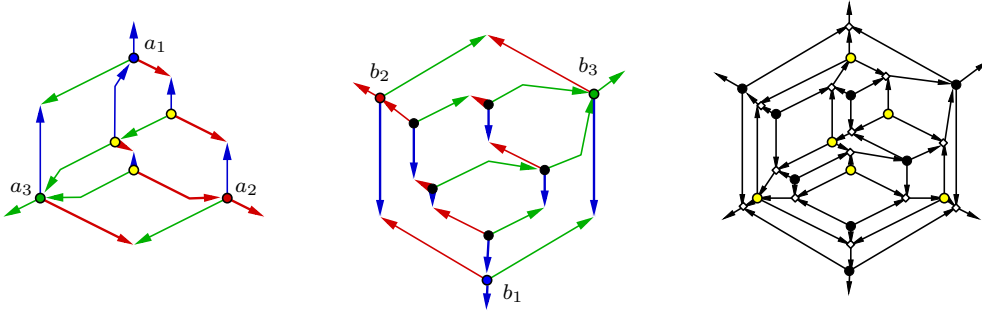


FIGURE 2.6 – Les bijections du Théorème 2.2

La structure de treillis des bois de Schnyder se comprend mieux sur les 3-orientations : soit X une 3-orientation de \widetilde{M}^σ et soit C un circuit de X . En retournant tous les arcs du circuit C on obtient un nouveau bois de Schnyder X^C . Si C est un circuit simple, alors soit il est orienté dans le sens anti-horaire, soit il est orienté dans le sens horaire. On définit la relation $X \succ X^C$ si C est un circuit horaire de X . La clôture transitive \succ^* définit une relation d'ordre sur les 3-orientations de \widetilde{M}^σ .

Théorème 2.3. *La relation \succ^* est la relation d'ordre d'un treillis distributif sur l'ensemble des 3-orientations de la suspension complétée \widetilde{M}^σ . L'unique 3-orientation minimale de \widetilde{M}^σ est celle qui ne contient aucun circuit horaire.*

Au vu du Théorème 2.2, une suspension M^σ admet un unique bois de Schnyder minimal S_{Min} . La Figure 2.7 montre les 2 sous-structures interdites dans S_{Min} :

- Une arête simplement orientée entrante en v et coloriée $i - 1$ telle que l'arête suivante dans le sens anti-horaire soit doublement orientée, sortante de v avec la couleur $i + 1$ et entrante avec la couleur i .
- Un triangle anti-horaire simplement orienté, utilisant les couleurs 1,2 et 3 dans le sens anti-horaire.

Un algorithme linéaire pour calculer S_{Min} a été décrit et analysé par Fusy et al. [FSP08] :

Théorème 2.4. [FSP08]

Soit M^σ une suspension. Le bois de Snyder minimal S_{Min} de M^σ peut-être calculé en temps linéaire.

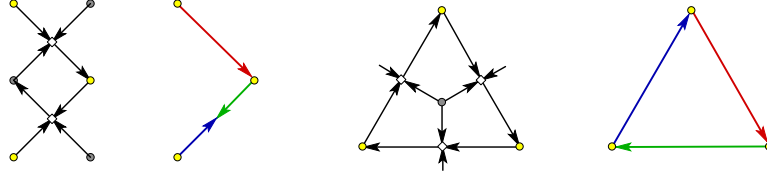


FIGURE 2.7 – Les 2 types de circuits absents des 3-orientations minimales ainsi que leurs équivalents pour les bois de Snyder.

2.2.4 Fusion et Division

Les opérations de *fusion* et *division* introduites dans cette section opèrent sur les bois de Snyder ainsi que sur leur carte sous-jacente. Ces opérations sont en fait l'inverse l'une de l'autre et correspondent à des suppressions ou des insertions d'arêtes dans la carte sous-jacente.

Un *angle* est un couple d'arêtes apparaissant successivement (dans le sens horaire) autour d'un même sommet. Étant donné un bois de Snyder, un *genou au sommet* v est un angle de v tel qu'une des arêtes est entrante pour v et l'autre est sortante.

Un genou peut être de 2 types : si l'arête entrante du genou est après l'arête sortante dans le sens anti-horaire on parle de *genou-cw* sinon on parle de *genou-ccw*.

Soit $(u, v), (v, w)$ un genou en v . Supposons que l'arête (v, w) soit de couleur i ; d'après la condition (W3) la couleur de (u, v) doit être $i + 1$ s'il s'agit d'un genou-cw et $i - 1$ s'il s'agit d'un genou-ccw. La *fusion du genou* consiste à supprimer l'arête sortante (v, w) et de rendre l'arête (u, v) doublement orientée, la demi-arête sortante de v devenant coloriée comme l'était l'arête (v, w) . Selon le type de genou on parle de fusion anti-horaire ou fusion horaire. La Figure 2.8 illustre cette définition.

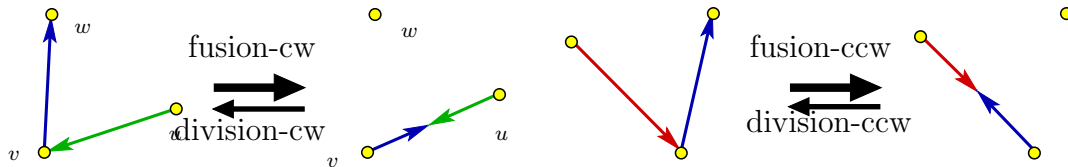


FIGURE 2.8 – Fusion et division anti-horaire et horaire.

Lemme 2.1. Soit S un bois de Snyder, la coloration et l'orientation des arêtes après la fusion d'un genou est encore un bois de Snyder.

Preuve Les conditions (W1), (W2) et (W3) des bois de Snyder restent vraies après l'opération de fusion. Au lieu de montrer la condition (W4) nous utilisons la bijection avec les 3-orientations (Théorème 2.2). La fusion d'un genou correspond à la suppression d'un sommet-arête comme le montre la Figure 2.9. Le résultat de la fusion est encore une 3-orientation. □

Une *division d'une arête double* est l'opération inverse d'une fusion. Cependant, une division n'est pas déterminée par le choix d'une arête double. Une arête double peut être divisée dans une et une seule de ses faces adjacentes (cette face est celle pointée par la demi-arête sortante du sommet arête dans la 3-orientation). Cela correspond à déterminer si la division est l'inverse d'une fusion-cw ou d'une fusion-ccw, c'est à dire d'une *division-cw* ou d'une *division-ccw*. Ce choix détermine la couleur des arêtes

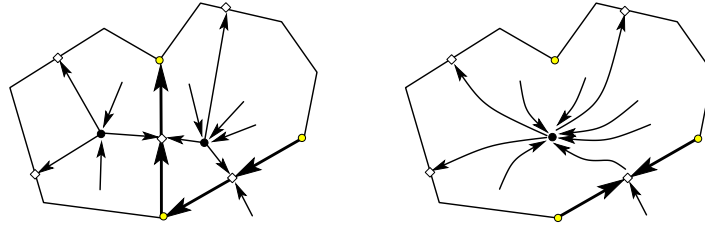


FIGURE 2.9 – Une fusion horaire dans une 3-orientation.

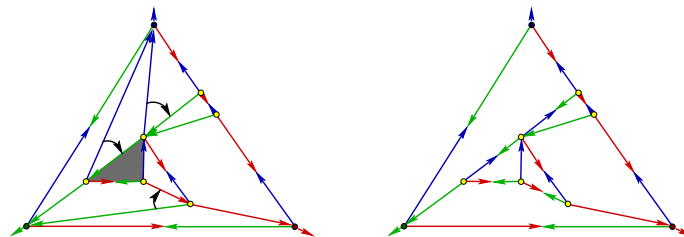
résultantes. Toutefois même si ce choix est fait, il reste encore à déterminer l'extrémité de l'arête créée par la division.

Dans notre contexte nous allons simplement considérer une version spécifique des divisions. Une *division-cw courte* est l'inverse d'une fusion-cw avec la condition supplémentaire que (u, w) soit une arête, i.e., que u, v, w forme un triangle.

2.3 L'algorithme de dessin convexe compact

Soit M^σ une carte 3-connexe avec n sommets et f faces. L'algorithme de dessin se décompose en 5 étapes :

- (A1) Choisir 3 sommets de la face externe pour obtenir une suspension M^σ .
- (A2) Calculer le bois de Schnyder minimal S_{Min} de M^σ . Soit $S_0 = S_{\text{Min}}$.
- (A3) Calculer une séquence maximale de fusion-cw $S_0 \rightarrow S_1 \rightarrow \dots \rightarrow S_k$ de bois de Schnyder, i.e. S_{i+1} est obtenu à partir de S_i par une fusion-cw de S_i et S_k ne contient pas de genou-cw.
- (A4) Appliquer l'algorithme de dessin par comptage de faces sur S_k . Celui produit un dessin convexe sur une grille $(f - k - 1) \times (f - k - 1)$.
- (A5) Insérer les arêtes supprimées pendant les opérations de fusion dans le dessin produit à l'étape précédente.

FIGURE 2.10 – Sur la gauche, le bois de Schnyder S_0 , les genoux-cw sont indiqués par des petites flèches. Sur la droite S_k , le dernier bois de Schnyder de la séquence.

La Figure 2.10 illustre l'étape (A3) de l'algorithme.

Notez que le bois de Schnyder obtenu peut devenir intérieurement 3-connexe.

Le triangle gris de la partie gauche de la Figure 2.10 contient un genou-cw qui disparaît lors d'une fusion-cw (cf. partie droite de la figure). Une propriété importante pour l'analyse de l'algorithme est qu'une fusion-cw ne fait jamais disparaître de genou-cw.

2.3.1 Le dessin est convexe

Théorème 2.5. *L'insertion des arêtes supprimées lors de la séquence de fusions-cw dans le dessin de S_k obtenu à l'étape (A4) laisse le dessin planaire et convexe.*

Les étapes (A4) et (A5) de l'algorithme sont illustrées dans la Figure 2.11. Les définitions et le lemme suivant sont essentiels pour la preuve de ce Théorème.

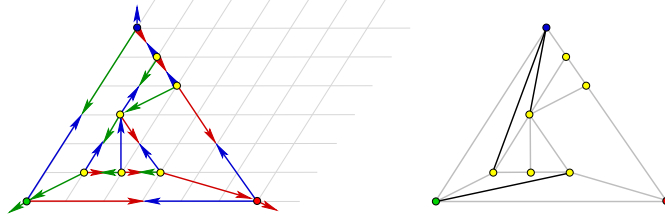


FIGURE 2.11 – Un exemple de graphe avec $n = f = 9$ dessiné sur une grille 6×6 .

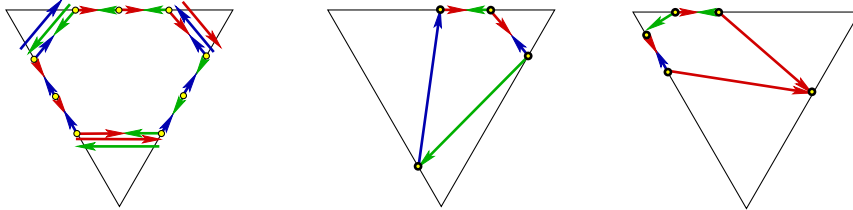


FIGURE 2.12 – La structure générique d'une face telle que décrite dans le Lemme 2.2 ainsi que de exemples concrets.

Soit f une face. Le type d'une arête par rapport à f est :

- $i-j$ si elle est bicoloriée et que la couleur i apparait avant la couleur j lorsqu'on fait le tour de la face dans le sens horaire.
- i -cw (resp. i -ccw) si elle est simplement orientée dans le sens horaire (resp. anti-horaire) et de couleur i .

Lemme 2.2. *Étant donné un bois de Schnyder d'une suspension M^σ et F une face interne de M^σ . L'orientation et la coloration des arêtes autour de F dans le sens horaire suivent le motif suivant (voir Figure 2.12) : Une arête de l'un de ces types $\{1$ -cw, 3 -ccw, 1 - $3\}$, puis des (potentiellement 0) arêtes 2 - 3 , puis une arête de type $\{2$ -cw, 1 -ccw, 2 - $1\}$, puis des (potentiellement 0) arêtes 3 - 1 puis une arête de type $\{3$ -cw, 2 -ccw, 3 - $2\}$, puis des (potentiellement 0) arêtes 1 - 2 .*

Preuve Il y a une bijection entre les bois de Schnyder de M^σ et ceux du dual M^{σ^*} (Théorème 2.2). Cette bijection peut être définie localement à l'aide des règles de la Figure 2.13.

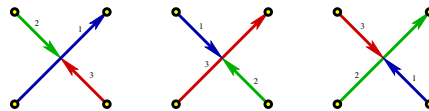


FIGURE 2.13 – Règles d'orientation et de coloration des arêtes duales.

A partir de cette règle, l'énoncé du lemme est équivalent à la condition (W3) sur le sommet v_F correspondant à la face F dans M^{σ^*} . □

Dans la Figure 2.12 les faces sont dessinées avec un triangle englobant. C'est l'une des propriétés des dessins par comptage de faces décrit dans la sous-section 2.2.2 : tous les sommets d'une face sont situés sur un tel triangle. La planarité du dessin nous garantit qu'il n'y a pas de sommets à l'intérieur de la face. En fait, il n'y a pas de sommets à l'intérieur du triangle englobant la face.

Lemme 2.3. *Soit S un bois de Schnyder d'une suspension M^σ .*

Soit $\mu(M)$ le dessin de M par comptage de faces dans le plan $x_1 + x_2 + x_3 = f - 1$ et F une face interne de M . Les sommets de F sont placés sur le bord d'un triangle dont les côtés sont $x_1 = c_1$, $x_2 = c_2$ et $x_3 = c_3$ pour certaines valeurs de c_1, c_2, c_3 (voir Figure 2.12).

Preuve Le Lemme 2.2 indique la couleur et l'orientation des arêtes autour de F . Le fait 4 et la Figure 2.4 indiquent comment les arêtes sont plongées autour d'un sommet en fonction de leur couleur et leur orientation. La preuve du lemme découle directement de la combinaison de ces deux observations. \square

Preuve du Théorème 2.5. Considérons une opération de fusion effectuée à l'étape (A3) de l'algorithme. L'arête doublement orientée e obtenue par l'opération de fusion est une arête de S_k . Supposons sans perte de généralité que l'arête $e = (u, v)$ était initialement coloriée 3 et que la fusion était une fusion-cw en v . Cela signifie que l'arête $e' = (v, w)$ qui a été fusionnée dans e était coloriée 1 (c'est exactement la situation décrite à gauche dans la Figure 2.8).

Dans le dessin de S_k considérons la face F qui est à gauche de l'arête e et considérons ∇ le triangle circonscrit de F (Lemme 2.3). A partir des couleurs de l'arête e , on a, grâce au Lemme 2.3, que v et u sont sur le bord du triangle ∇ d'équation $x_2 = c_2$. De plus, $u_1 > v_1$. Le sommet w appartient aussi à F . L'arête e' supprimée par l'opération de fusion était entrante en w entre l'arête sortante 2 et l'arête sortante 3 dans le sens horaire. Donc, F est contenu dans la région $R_1(w)$.

Supposons que la surface de F est dans la région $R_1(w)$ par rapport au bois de Schnyder final S_k obtenu par la séquence de fusions. Dans ce cas, le sommet w est placé sur le bord de ∇ d'équation $x_1 = c_1$. La position de v et w dans le triangle ∇ implique que le segment correspondant à l'arête (v, w) peut être ajouté au dessin sans générer de croisements.

Supposons maintenant que F ne reste pas dans la région $R_1(w)$ par rapport au bois de Schnyder final S_k . Ceci peut se produire uniquement si lors d'une fusion ultérieure d'un genou de w porté par face F . Comme il s'agit d'une fusion-cw, cette opération fusionne une arête coloriée 2 sortante de w avec une arête entrante 1. Les autres fusions ne changent rien à la situation. Dans ce cas, le sommet w est placé sur le bord de ∇ d'équation $x_3 = c_3$. A partir des positions de v et de w sur le triangle ∇ , nous pouvons de nouveau conclure que le segment (v, w) peut-être ajouté au dessin sans générer de croisements.

Une face F d'un dessin S_k peut également être le siège de plusieurs réinsertions d'arêtes fusionnées. Nous prétendons que toutes ces réinsertions peuvent se faire sans conflit : la bordure de F est un cycle C de la carte M . Les arêtes qui ont disparu lors de la transformation de S_0 en S_k sont des cordes du cycle C . Dans M , toutes ces cordes peuvent être dessinées sans croisement à l'intérieur de C . Dans le dessin de S_k , le cycle C est convexe. Donc toutes ces cordes peuvent être réinsérées sans générer de croisements.

Il reste juste à remarquer que l'ajout d'une corde à une face convexe engendre deux faces convexes. Le dessin final est donc lui aussi convexe. \square

2.3.2 Nombre de fusions

La taille de la grille utilisée par notre algorithme de dessin dépend de la longueur k de la séquence de fusions de l'étape (A3). Dans cette sous-section nous allons donner une borne inférieure pour k . Cette borne dépend d'un paramètre naturel sur le bois de Schnyder minimal calculé à l'étape (A2) de l'algorithme.

Pour s'échauffer considérons le cas où M est une triangulation et S est le bois de Schnyder minimal de M . Considérons les $(2n - 4) - 4$ faces contenant que des arêtes simplement orientées (seules les arêtes

externes sont doublement orientées). Ces triangles peuvent être partitionnés en 2 classes : ceux ayant au moins 2 arêtes dans le sens horaire (en faisant le tour de la face) et ceux ayant au moins 2 arêtes dans le sens anti-horaire. Soit C_1 et C_2 le nombre de faces dans chacune de ces classes.

Supposons qu'il y a plus de triangles dans la première classe que dans la deuxième : $C_1 \geq n - 4 \geq C_2$. Dans un triangle T de la première classe, il y a un genou fait de 2 arêtes horaires consécutives de T . Ce genou est un candidat pour une fusion horaire. Comme une arête n'est horaire que dans un seul de ses triangles adjacents, ces C_1 fusions peuvent être faites indépendamment. Partant de S , il existe une séquence de fusion de longueur $k \geq C_1 \geq n - 4$. Grâce à cette estimation on peut dessiner une triangulation sur une grille de taille au plus $(f - (n - 4) - 1) \times (f - (n - 4) - 1) = (n - 1) \times (n - 1)$.

Considérons à nouveau une triangulation M d'un bois de Schnyder S et essayons de donner une estimation plus précise du nombre de fusions applicables à S . Considérons un genou-cw en v . soit i la couleur de l'arête entrante (u, v) du genou. L'arête (u, v) montre que v est un noeud interne de l'arbre T_i . Le nombre de genoux-cw est $\sum_i \text{inner}(T_i)$. Si une arête simplement orientée (v_1, v_2) participe à 2 genoux-cw différents, alors v_1 et v_2 sont tous les deux des sommets de genoux-cw. Cela implique que le triangle à droite de l'arête (v_1, v_2) est un triangle horaire. Un triangle horaire contribue à 3 genoux-cw qui sont deux à deux incompatibles. Si Δ_S^{\rightarrow} est le nombre de triangles horaires de S alors le nombre de fusions horaires qui peuvent être faites simultanément sur S est au moins $\sum_i \text{inner}(T_i) - 2\Delta_S^{\rightarrow}$. De plus nous avons prouvé dans [BLM02]³ que

$$\sum_i \text{inner}(T_i) = n - 4 + \Delta_S^{\rightarrow} + \Delta_S^{\leftarrow}$$

Cela nous amène à la Proposition 2.1.

Proposition 2.1. *Soit S un bois de Schnyder. Soit Δ_S^{\rightarrow} (resp. Δ_S^{\leftarrow}) le nombre de triangles horaires (resp. anti-horaires) de S .*

Le nombre de fusions horaires applicables dans une séquence de fusions partant de S est au moins $n - 4 - \Delta_S^{\rightarrow} + \Delta_S^{\leftarrow}$.

En combinant le Théorème 2.5 et la Proposition 2.1 on trouve qu'une triangulation ayant un bois de Schnyder s peut être dessinée sur une grille $(n - 1 + \Delta_S^{\rightarrow} - \Delta_S^{\leftarrow}) \times (n - 1 + \Delta_S^{\rightarrow} - \Delta_S^{\leftarrow})$. Il est intéressant de remarquer qu'un cas spécial de cette borne (Corollaire 2.1) a été obtenu par Zhang et He [ZH03] avec une technique différente.

Corollaire 2.1. *Une triangulation M à n sommets admet un dessin en lignes droites sur une grille de taille $(n - 1 - \Delta_{S_{\text{Min}}}^{\leftarrow}) \times (n - 1 - \Delta_{S_{\text{Min}}}^{\leftarrow})$. où S_{Min} est le bois de Schnyder Minimal de M .*

Intéressons-nous maintenant au cas plus général des bois de Schnyder 3-connexes. Pour un tel bois S nous introduisons 2 paramètres :

- Δ_S^{\leftarrow} le nombre de faces, avec une arête anti-horaire pour chacune des 3 couleurs (ces arêtes n'étant pas forcément simplement orientées).
- $\Delta_S^{\overleftarrow{\Delta}}$ le nombre de triangles horaires constitués d'arêtes simplement orientées plus le motif suivant : une arête simplement orientée entrante en v de couleur $i - 1$ telle que l'arête suivante dans le sens anti-horaire autour de v est doublement orientée, sortante en v de couleur $i + 1$ et entrante de couleur i : cf. Figure 2.14.

Théorème 2.6. *Soit S le bois de Schnyder d'une suspension. Le nombre de fusions-cw qui peuvent être appliquées à S est d'au moins $f - n + \Delta^{\leftarrow}(S) - \Delta_S^{\overleftarrow{\Delta}}$.*

Preuve Dans un premier temps nous allons considérer le cas plus simple où la face externe est un triangle.

La borne obtenue de la manière suivante à partir de borne de la Proposition 2.1 : partant d'un bois de Schnyder S on construit un bois de Schnyder S' d'une triangulation tel que S est obtenue à partir

3. Une preuve plus simple a ensuite été proposée dans [LLS04].

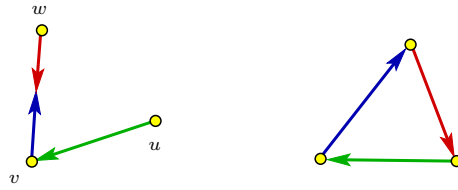


FIGURE 2.14 – Les 2 motifs qui comptent pour $\overline{\Delta^+}$.

de S' par une séquence de k fusions-cw. La Proposition 2.1 nous donne une borne k' sur le nombre de fusions-cw applicables à S' . La différence $k' - k$ nous donne la borne sur le nombre de fusions-cw applicables à S .

Considérons une face interne F de S . Sur la bordure de F il y a 3 arêtes spéciales, qui sont les duales des arêtes sortantes du sommet dual v_F . Ces arêtes spéciales sont séparées par des chemins doublement orientés (voir Lemme 2.2). Les arêtes des chemins doublement orientés de F de couleurs $i, i + 1$ sont divisées de sorte que l'arête coloriée $i - 1$ ait pour destination le dernier point du chemin doublement orienté colorié $i + 1, i$. Ceci peut être obtenu par une séquence de divisions-cw (voir Figure 2.15).

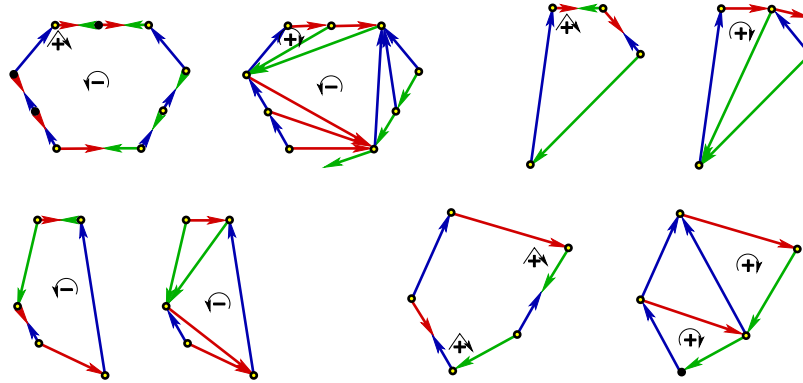


FIGURE 2.15 – Quatre exemples de triangulations d'une face.

En appliquant cette construction à toutes les faces internes de S on obtient un bois de Schnyder S' d'une triangulation.

Les observations suivantes sont cruciales :

- Le nombre de fusions-cw applicables à S' est $k' \geq n - 4 + \Delta_{S'}^{\oplus} - \Delta_{S'}^{\ominus}$ (Proposition 2.1).
- Le bois de Schnyder original S peut-être obtenu à partir de S' par une séquence de k fusions-cw.
- Chaque fusion réduit le nombre de faces de 1 et donc $k = (2n - 4) - f$.
- $\Delta_{S'}^{\oplus} = \Delta_S^{\oplus}$ et $\Delta_{S'}^{\ominus} = \Delta_S^{\ominus}$ (c.f. Figure 2.15).

Ainsi le nombre de fusions-cw applicables à S est $k' - k \geq f - n + \Delta_S^{\oplus} - \Delta_S^{\ominus}$.

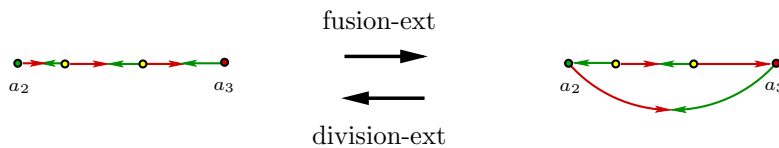


FIGURE 2.16 – Fusion et division externes.

Maintenant supposons que la face externe contienne plus de 3 sommets. Partant de S nous produisons un bois de Schnyder S^* avec une face externe triangulaire. Ceci peut être fait à l'aide de *divisions externes*, comme le montre la Figure 2.16. Le nombre de divisions externes pour passer de S à S^* est $t \in \{0, 1, 2, 3\}$.

Nous savons déjà que S^* admet $k^* \geq f^* - n + \Delta_{S^*}^{\oplus} - \Delta_{S^*}^{\ominus}$ fusions-cw.

En comptant la variation du nombre de faces et de genoux-cw entre S et S^* on montre que là encore S admet au moins $f - n + \Delta_S^{\oplus} - \Delta_S^{\ominus}$ fusions-cw. \square

Étant donné un bois de Schnyder arbitraire, la contribution $\Delta^{\oplus} - \Delta^{\ominus}$ peut-être négative. Cependant en prenant $S = S_{\text{Min}}$ nous sommes certains que cette contribution est positive. Nous pouvons résumer cette section par le Théorème suivant :

Théorème 2.7. *Toute carte plane 3-connexe M à n sommets admet un dessin convexe sur une grille de taille $(n - 1 - \Delta_{S_{\text{Min}}}^{\oplus}) \times (n - 1 - \Delta_{S_{\text{Min}}}^{\ominus})$, où $\Delta_{S_{\text{Min}}}^{\oplus} \geq 0$ est le nombre de faces ayant une arête anti-horaire dans chaque couleur dans S_{Min} . Un tel dessin peut-être obtenu en temps linéaire.*

2.4 Améliorations et Limitations

Notre ambition était de concevoir un algorithme de dessin convexe pour les cartes 3-connexes qui améliore les meilleurs algorithmes pour cette tâche dans tous les cas. Or il existe des triangulations pour lesquelles le paramètre $\Delta_{S_{\text{Min}}}^{\oplus} = 0$. Ce sont les triangulations piles (celles qui peuvent être épluchées par suppression récursive des sommets de degré 3). La Figure 2.17 montre un exemple d'une telle triangulation. Pour ces triangulations notre algorithme nécessite une grille de taille $(n - 1) \times (n - 1)$ alors que l'algorithme de Schnyder utilise une grille de taille $(n - 2) \times (n - 2)$.

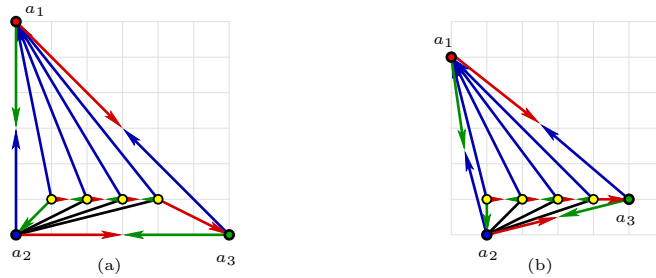


FIGURE 2.17 – (a) Une triangulation pile sur une grille $(n - 1) \times (n - 1)$. (b) La même carte dessinée avec la méthode améliorée.

Dans l'algorithme standard par comptage de faces, le sommet a_1 est en position $(f - k - 1, 0, 0)$, a_2 est en position $(0, f - k - 1, 0)$ et a_3 est en position $(0, 0, f - k - 1)$. Dans le cas où a_1, a_2 et a_3 sont choisis de sorte que a_1 soit voisin de a_2 et a_3 , il est possible de placer ces 3 sommets aux positions suivantes en gardant un dessin planaire et convexe : $a_1 \rightarrow (f - k - 2, 0, 1)$, $a_2 \rightarrow (1, f - k - 2, 0)$ et $a_3 \rightarrow (0, 1, f - k - 2)$.

En faisant ainsi on diminue d'une unité la largeur et la hauteur de la grille.

Ceci nous amène au résultat suivant :

Théorème 2.8. *Toute carte plane 3-connexe M à n sommets admet un dessin convexe sur une grille de taille $(n - 2 - \Delta_{S_{\text{Min}}}^{\oplus}) \times (n - 2 - \Delta_{S_{\text{Min}}}^{\ominus})$, où $\Delta_{S_{\text{Min}}}^{\oplus} \geq 0$ est le nombre de faces ayant une arête anti-horaire dans chaque couleur dans le bois de Schnyder minimal S_{Min} .*

Nous concluons cette section avec une application du Théorème 2.8 aux cartes intérieurement 3-connexes.

Une carte M est *intérieurement 3-connexe* si l'ajout d'un sommet v^+ relié à tous les sommets de la face externe donne une carte M^+ 3-connexe. Thomassen [Tho80] a prouvé que M possède un dessin (strictement) convexe si et seulement si M est intérieurement 3-connexe.

Un dessin est dit *intérieurement convexe* si toutes les faces internes sont convexes. Chrobak et Kant [CK97] ont adapté leur algorithme pour dessiner de manière convexe les graphes intérieurement 3-connexes sur des grilles de taille $(n-1) \times (n-2)$. Avec notre approche nous pouvons réduire cette borne.

Soit M une carte intérieurement 3-connexe à n sommets. Soit M^+ la carte obtenue à partir de M en ajoutant un sommet v^+ comme décrit ci-dessus. Soit $S = S_{\text{Min}}$ le bois de Schnyder minimal de la suspension M^+ avec $v^+ = a_1$. Comme M^+ possède $n+1$ sommets, le Théorème 2.8 nous garantit un dessin de M^+ sur la grille $(n-1-\Delta_S^{\oplus}) \times (n-1-\Delta_S^{\oplus})$. Comme la face externe de M^+ est un triangle, a_1 est le seul sommet sur la dernière ligne et la première colonne de la grille. Donc en supprimant a_1 du dessin on obtient :

Corollaire 2.2. *Une carte intérieurement 3-connexe M à n sommets peut être dessinée de manière intérieurement convexe sur une grille $(n-2-\Delta_S^{\oplus}) \times (n-2-\Delta_S^{\oplus})$, Où S est le bois de Schnyder minimal de M^+ .*

2.4.1 Etude en moyenne

Nous avons vu que dans le pire des cas, notre algorithme fait aussi bien que l'algorithme de Schnyder et dans le pire des cas on atteint la même borne que Schnyder $(n-2) \times (n-2)$ (dans le cas des triangulations piles, dont un exemple est donné dans la Figure 2.17)

Dans le cas des triangulations nous pouvons compléter ce résultat (dans le pire des cas) par un résultat en moyenne :

Théorème 2.9. [BGH⁺06] *L'algorithme du Théorème 2.8 produit des dessins convexe sur des grilles de taille $(7n/8 + o(n)) \times (7n/8 + o(n))$ en moyenne pour une triangulation à n sommets.*

Preuve Pour montrer cela, nous utilisons la bijection présentée dans [PS03] entre les bois de Schnyder minimaux à n sommets et une famille d'arbres dit équilibrés. Grâce à cette bijection nous pouvons estimer que le nombre moyen de feuilles l_i dans chacun des 3 arbres T_i est $5n/8 + o(n)$. De plus le paramètre Δ_S^{\oplus} vérifie la relation suivante [BLM02] pour tout bois de Schnyder S à n sommets :

$$\Delta_S^{\oplus} + l_1 + l_2 + l_3 = 2n - 5.$$

On en déduit que $\Delta_S^{\oplus} = n/8 + o(n)$. □

A l'autre extrême on trouve les cartes cubiques. Pour une carte cubique, toutes les arêtes sauf 3 sont doublement orientées. Il n'est donc pas possible d'effectuer des fusions et notre algorithme donne des dessins de même taille que l'algorithme classique par comptage de faces.

Entre ces 2 cas extrêmes se trouvent les cartes planaires 3-connexes. Nous nous sommes intéressés aux cartes planaires aléatoires à m arêtes. Pour générer aléatoirement ces cartes nous avons utilisé le générateur aléatoire de Schaeffer [Sch99]. Les résultats obtenus en faisant varier la valeur de m sont présentés dans la Figure 2.18. Remarquons tout d'abord que pour ces cartes, le nombre moyen de sommets et le nombre moyen de faces sont tous les deux égaux à $(m-2)/2$ pour des raisons de dualité. La Figure 2.18 montre une forte concentration autour de cette valeur moyenne. On observe également que la valeur moyenne du paramètre $\Delta_{S_{\text{Min}}}^{\oplus}$ est proche de $m/8$. Par conséquent, la taille moyenne des grilles utilisées pour ces cartes est autour de $3n/4 \times 3n/4$ contre $n \times n$ pour l'algorithme classique par comptage de faces.

2.5 Conclusion et Perspectives

Nous venons de voir comment les bois de Schnyder 3-connexe permettaient d'avoir des dessins de graphes convexe et compacts. Nous avons implémenté cet algorithme et il est maintenant intégré dans

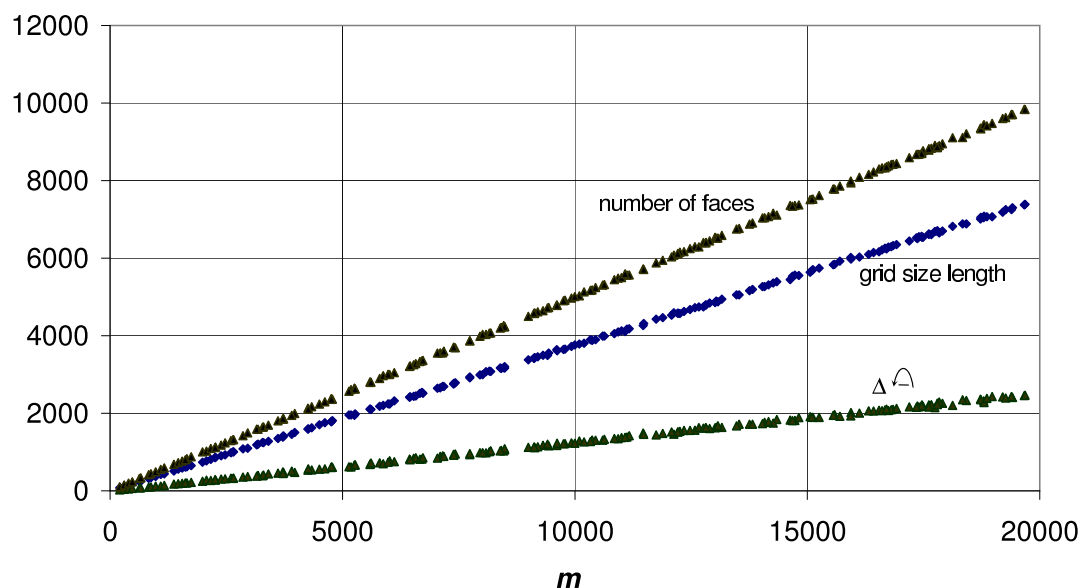


FIGURE 2.18 – Résultats expérimentaux sur des cartes 3-connexes aléatoires uniformes à m arêtes, pour des valeurs différentes de m .

la bibliothèque de dessin de graphes PIGALE⁴ développée par Hubert de Fraysseix et Patrice Ossona de Mendez.

Malgré cette avancée en terme de taille de grille pour la plupart des graphes, il n'existe pas d'algorithme pouvant dessiner tous les graphes planaires à n sommets sur des grilles plus petites que $(n - 2) \times (n - 2)$. On sait qu'il existe des cartes qui nécessitent des grilles de taille $(2n/3) \times (2n/3)$ (cartes formées de triangles imbriqués). Depuis 1989, l'écart entre ces 2 bornes n'a pas évolué. L'utilisation des bois de Schnyder pour la conception d'algorithmes (ou de contre exemples) pour rapprocher ces bornes est loin d'être évidente. Il faudrait explorer des pistes plus exotiques, c'est à dire qui ne soit pas basée sur les bois de Schnyder ou sur les ordres canoniques.

Une perspective plus évidente de travaux présentés dans ce chapitre serait l'énumération des bois de Schnyder 3-connexes. Alors que l'énumération des bois de Schnyder des cartes triangulées est maintenant établi [Bon05], l'énumération des bois de Schnyder 3-connexes reste ouverte. Une piste pour obtenir cette énumération serait de déterminer pour chaque bois de Schnyder maximum, combien de bois de Schnyder 3-connexes peuvent être obtenus à l'aide de fusions-cw.

Récemment, des généralisations des bois de Schnyder ont été proposées pour les cartes de genre supérieur. La première généralisation a été proposée par Castelli-Aleardi, Fusy et Lewiner [CAFL08]. Elle a l'avantage de fonctionner pour les cartes de genre arbitraire mais l'un des arbres et certaines arêtes jouent des rôles particuliers. La deuxième généralisation proposée par Gonçalves et Lévêque [GL12] (cf. Figure 2.19) quant à elle ne fonctionne que pour le genre 1, mais tous les sommets vérifient la condition locale. Les opérations de fusions et de divisions semblent s'étendre naturellement à ces objets. Ces opérations pourraient s'avérer être un outil précieux pour montrer de nouvelles propriétés de ces bois de Schnyder de genre 1.

4. <http://pigale.sourceforge.net>

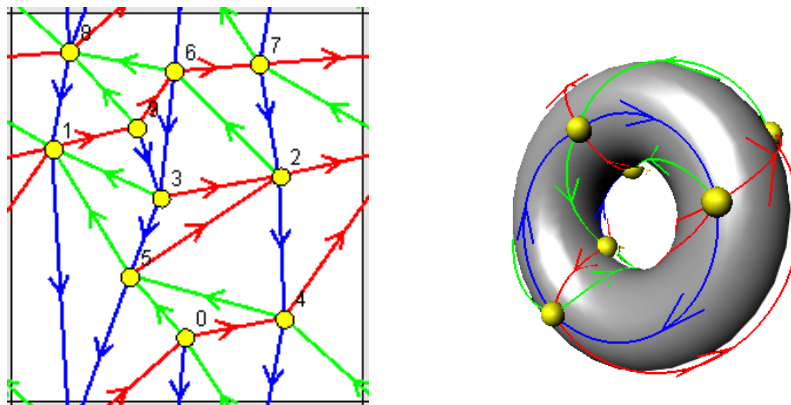


FIGURE 2.19 – Bois de Schnyder torique.

Chapitre 3

Graphes couvrants géométriques

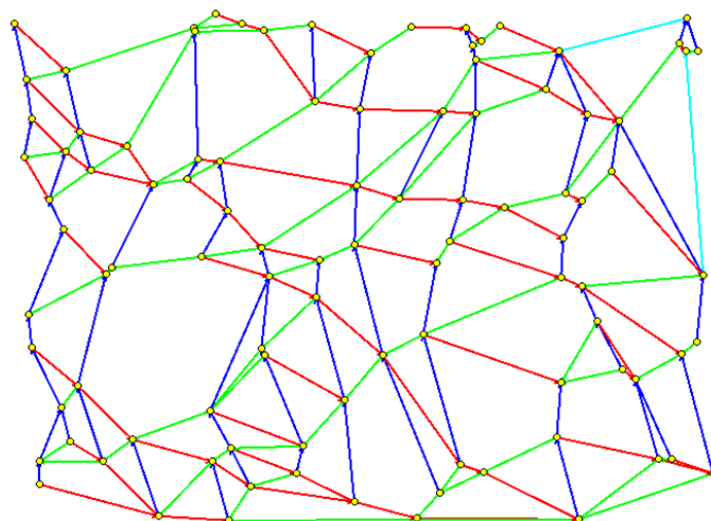


FIGURE 3.1 – Exemple de 6-spanner planaire de degré maximal 6.

Dans le chapitre précédent, nous avons vu comment, plonger une carte planaire dans le plan sans croisement et de sorte que les faces internes soit convexes. En d'autres termes : étant donné un ensemble d'arêtes, le problème est de déterminer les coordonnées des sommets pour obtenir un plongement ayant de bonnes propriétés. Dans ce chapitre nous allons nous intéresser au problème dual dans un certain sens : étant donné un nuage de points S dans le plan, trouver l'ensemble des arêtes formant un graphe connexe couvrant tous les sommets tel que pour toute paire de points, la distance dans le graphe soit au plus t fois la distance dans le plan. On dit que le graphe ainsi construit est un t -spanner. Toute la difficulté consiste à utiliser le moins d'arêtes possible mais aussi à tenir compte d'autres contraintes selon le contexte. Par exemple : minimiser le nombre d'arêtes ou la somme des longueurs des arêtes, minimiser le degré maximum du graphe, interdire les croisements d'arêtes, minimiser la longueur maximale des arêtes utilisées... Le lecteur pourra se reporter à l'ouvrage de Narasimhan et Smid [NS07] pour mesurer l'étendue du domaine.

Les contributions au domaine présentées ici sont le fruit de collaborations avec Ljubomir Perković (de Paul University - Chicago), Nicolas Hanusse, David Ilcinkas et Cyril Gavoille [BGHI10a, BGHP10].

3.1 Introduction

Un graphe géométrique est un graphe valué dont l'ensemble des sommets S est un ensemble de points du plan (ou plus généralement des points d'un espace métrique quelconque) et où les arêtes sont des segments joignant ces sommets. Le *poids* d'une arête est donné par la distance Euclidienne entre ses extrémités.

Les graphes géométriques modélisent naturellement de nombreux problèmes pratiques intervenant dans différents domaines de l'informatique allant des réseaux (réseaux maillés, réseaux ad hoc) à la géométrie algorithmique. Parmi les graphes géométriques célèbres on peut citer les triangulations de Delaunay, les graphes de Yao, les graphes de Gabriel [GO97]. Un des paramètres essentiels d'un graphe géométrique qui fait l'objet d'un nombre considérable de travaux est l'*étirement*. Étant donné un sous graphe H de G , l'*étirement* de H est

$$\max_{(u,v) \in S^2} \frac{d_H(u,v)}{d_G(u,v)}.$$

On dit qu'un graphe est un t -*spanner* si son étirement est au plus t . Dans le cas des graphes géométriques, le graphe que l'on souhaite le plus souvent approximer est le graphe complet (Euclidien) (noté \mathcal{E}).

Parmi les spanners géométriques les plus populaires on peut citer les Θ_k -graphes [Cla87, Kei88]. Ces graphes sont définis de la manière suivante : autour de chaque point p de S on partitionne l'espace en $k \geq 2$ cônes réguliers de sommet p ; un point $q \neq p$ d'un cône C est adjacent à p si et seulement si q est le point le plus "proche" de p dans C , c'est à dire le point de C ayant le plus proche projeté orthogonal sur la bissectrice de C .

Les Θ_k -graphes sont connus pour être de bons spanners [RS91] : pour $k > 6$, Θ_k est un $\frac{1}{(1-2 \sin(\pi/k))}$ -spanner. Par exemple Θ_7 est un 7.562-spanner. De plus ces graphes ont deux propriétés supplémentaires intéressantes : ils ont peu d'arêtes (au plus kn) et ils peuvent être construits de manière distribuée.

Lorsqu'on travaille avec des graphes géométriques, la planarité est également une propriété naturelle et importante. Dans le cas de réseau ad hoc, certains algorithmes de routage [BMSU01] se basent sur la planarité du graphe sous-jacent.

Chew [Che86] fut le premier à montrer qu'il existait un t -spanner planaire, c'est à dire sans croisement d'arêtes, pour une certaine valeur de t . Plus précisément, il montra que les triangulations de L_1 -Delaunay (*i.e.* triangulations de Delaunay définies en utilisant la distance L_1 à la place de la distance Euclidienne) étaient des $\sqrt{10}$ -spanners. Plus tard [Che89] il améliora ce résultat en montrant qu'il existait des 2-spanners planaires.

Rapidement la question de l'étirement des triangulations de Delaunay s'est posée. La première borne supérieure sera établie par Dobkin et al. [DFS90], en montrant que les triangulations de Delaunay sont des 5.08-spanners. Puis cette borne descendit à 2.41 [KG92]. Très récemment cette borne a été améliorée à 1.998 [Xia11], faisant ainsi des triangulations de Delaunay les meilleurs spanners planaires connus. Côté borne inférieure, il est facile de construire une triangulation de Delaunay d'étirement arbitrairement proche de $\pi/2$. Ce contre-exemple s'obtient en plaçant les points de S quasiment sur un cercle de sorte qu'il existe une paire de points diamétralement opposés dont le plus court chemin s'approche d'un demi-cercle. Longtemps les gens ont pensé que les triangulations de Delaunay étaient des $\pi/2$ -spanners. Mais depuis, Bose et al. [BDL⁺11] ont montré que $\pi/2$ n'était pas la bonne borne en construisant des contre-exemples d'étirement 1.5846 ($\pi/2 \approx 1.5707$). La meilleure borne inférieure connue est actuellement de 1.5932 [XZ11].

Dans le contexte des réseaux ad hoc, où les noeuds ont des ressources limitées, borner le nombre de liens de communication que doit maintenir chaque noeud devient essentiel. C'est pourquoi la construction de spanners (planaires) de degré borné est rapidement apparue comme une question importante. Pour illustrer cela, considérons les réseaux *scatternets* [LSW04] basés sur la technologie Bluetooth. Un scatternet est la superposition de réseaux Bluetooth (également appelés *piconets*) constitués d'un maître et d'au plus 7 esclaves. Chaque noeud peut appartenir à plusieurs piconets, mais il ne peut pas être simultanément maître de plusieurs piconets. Choisir quels sont les sommets qui doivent être des maîtres

papier	Δ	étirement
Dobkin <i>et al.</i> [ADDJ90]	∞	$\frac{\pi(1+\sqrt{5})}{2} \approx 5.08$
Keil & Gutwin [KG92]	∞	$\frac{4\pi\sqrt{3}}{9} \approx 2.42$
Xia [Xia11]	∞	$C_0 \approx 1.998$
Chew [Che89]	∞	2
Bose <i>et al.</i> [BGS05]	27	$(\pi + 1)C_0$
Li & Wang [LW04]	23	$(1 + \pi \sin \frac{\pi}{4})C_0$
Bose <i>et al.</i> [BSX09]	17	$(2 + 2\sqrt{3} + \frac{3\pi}{2} + 2\pi \sin(\frac{\pi}{12}))C_0$
Kanj & Perković [KP08]	14	$(1 + \frac{2\pi}{14 \cos(\frac{\pi}{14})})C_0$
Kanj & Xia [KX12]	11	< 7
Bose, Carmi & Chaitman-Yerushalmi [BCCY12]	7	$(1 + \sqrt{2})^2 C_0$
Bose, Carmi & Chaitman-Yerushalmi [BCCY12]	6	$\frac{1}{1 - \tan(\pi/7)(1 + 1/\cos(\pi/14))} C_0$
Section 3.6	9	6
Section 3.7	6	6

TABLE 3.1 – Résultats sur les spanners planaires. C_0 désigne la meilleure borne connue sur l'étirement des triangulations de Delaunay.

devient problématique si on part d'un réseau de degré non borné ou borné par une valeur supérieure à 7. En revanche lorsqu'on part d'un réseau (un spanner) de degré au plus 7, il suffit de prendre comme ensemble de maîtres n'importe quel ensemble dominant (par exemple un ensemble indépendant maximal). Pour cette dernière opération il existe de nombreux algorithmes distribués efficaces [Lub85a].

Bose *et al.* [BGS05] furent les premiers à proposer un spanner planaire de degré borné. Le degré maximal et l'étirement qu'ils ont obtenu fut amélioré par Li et Wang [LW04], Bose *et al.* [BSX09], puis par Kanj et Perković [KP08] (Voir le tableau 3.1 pour toutes ces bornes).

Contributions Notre première contribution [BGHI10a] est une unification de 3 concepts : les triangulations de TD-Delaunay étudiées par Chew [Che86], les plongements géodésiques coplanaires [Mil02, FZ08] (structures en lien étroit avec les bois de Schnyder) et les demi Θ_6 -graphes (graphes définis en ne considérant qu'un cône sur deux dans la construction du Θ_6 -graphe). Plus précisément, étant donné un nuage de points S en position générale¹, la triangulation de TD-Delaunay de S , le plongement géodésique de S et le demi Θ_6 -graphe de S sont égaux.

De cette de unification nous en déduisons 2 résultats secondaires :

1. Θ_6 est l'union disjointe de deux triangulations de TD-Delaunay, et donc Θ_6 est un 2-spanner. Rappelons qu'on ignorait si Θ_6 était un spanner ou non.
2. Toute triangulation est TD-Delaunay réalisable. Cela signifie que pour toute carte planaire maximale G , il existe un nuage de points S tel que la triangulation de TD-Delaunay de S soit G . Rappelons que décider si une triangulation (vue comme une carte combinatoire) est Delaunay réalisable est un problème qui est resté longtemps ouvert [Dil90] et qu'il existe maintenant des algorithmes polynomiaux pour ce problème [HMS00], mais il n'existe toujours pas de caractérisation simple de ces graphes.

Notre deuxième contribution [BGHP10] est de montrer qu'il existe des 6-spanners planaires de degré maximum 6, améliorant ainsi une longue série d'articles sur les spanners planaires de degré maximal borné. Les précédents résultats avaient tous comme première étape le calcul d'une triangulation de Delaunay. La construction que nous avons proposée est quant à elle basée sur les triangulations de TD-Delaunay et l'analyse de cette construction s'appuie très fortement sur la coloration et l'orientation des arêtes de la triangulation de TD-Delaunay.

1. il n'y a pas de paire de points de S définissant une pente de 0, $\pi/3$ ou $-\pi/3$.

3.2 Unification

3.2.1 Demi Θ_6 -graphe

Un *cône* est une région du plan comprise entre 2 rayons partant d'un même point appelé *apex*. Pour chaque cône C , on note ℓ_C la bissectrice du cône, et pour chaque point p , soit $C^p = \{x + p : x \in C\}$.

Considérons les rayons obtenus par une rotation horaire de la partie positive de l'axe des x par un angle $2i\pi/k$ pour chaque entier $i < k$. Chaque paire de rayons consécutifs $2(i-1)\pi/k$ et $2i\pi/k$ définit un cône, dénoté par A_i , dont l'apex est l'origine. Soit $\mathcal{A}_k = \{A_1, \dots, A_k\}$.

Le Θ_k -graphe orienté d'un ensemble de point $S \subset \mathbb{R}^2$, dénoté par $\overrightarrow{\Theta}_k(S)$ est défini de la manière suivante :

1. l'ensemble des sommets de $\overrightarrow{\Theta}_k(S)$ est S ; et
2. (p, r) est un arc de $\overrightarrow{\Theta}_k(S)$ si et seulement si il y a un cône $A_i \in \mathcal{A}_k$ tel que $r \in A_i^p \setminus \{p\}$ le point de $S \cup A_i^p \setminus \{p\}$ ayant le projeté orthogonal sur ℓ_C^p est le plus proche de p .

Cette définition ne fait pas d'hypothèse particulière sur la position relative des points de S . En particulier, il se peut que dans $\overrightarrow{\Theta}_k(S)$ un cône contienne plusieurs arcs de même longueur (induisant un degré sortant supérieur à k), ou bien qu'un arc soit situé sur le bord d'un cône. Nous considérons ici, que les points de S sont en *position générale*, c'est à dire qu'il n'existe pas de paire de points de S alignés avec un des axes des cônes. La notion de "position générale" est discutée plus en détail dans [BGHI10a].

Nous introduisons maintenant un nouveau graphe, appelé *demi- Θ_k -graphe*, défini de la manière suivante :

Définition 3.1. *Le graphe orienté demi- Θ_k -graphe d'un nuage de points $S \subset \mathbb{R}^2$, noté $\frac{1}{2}\overrightarrow{\Theta}_k(S)$, est le graphe orienté induit par tous les arcs (p, r) de $\overrightarrow{\Theta}_k(S)$ tels que $r \in A_{2i}^p$ pour un nombre $i \in \{1, 2, 3\}$. On notera $\text{parent}_i(p) = r$.*

Quand $k \equiv 2 \pmod{4}$, nous dénotons par C_i le cône A_{2i} , et par \overline{C}_i le cône opposé de C_i , i.e., $\overline{C}_i = A_{2i+k/2 \pmod{6}}$ (observez que $2i + k/2$ est impair). Un arc (p, r) tel que $r \in C_i^p$ est dit *colorié i* .

Par la suite nous allons nous focaliser sur le graphe demi- Θ_6 . Ainsi, dans le sens anti-horaire en partant de l'axe positif des x , les 6 cônes de \mathcal{A}_6 sont rencontrés dans l'ordre $\overline{C}_2, C_1, \overline{C}_3, C_2, \overline{C}_1, C_3$ (voir Figure 3.2(a)). Les cônes C_1, C_2, C_3 sont dits *positifs* et les cônes $\overline{C}_1, \overline{C}_2, \overline{C}_3$ sont dits *négatifs*.

La Figure 3.2(b) montre un exemple de graphe demi- Θ_6 orienté sur 8 points.

3.2.2 Plongement géodésique

Soit \mathcal{P} le plan équipé de la base standard $(\mathbf{e}_x, \mathbf{e}_y)$, et soit S un nuage de points fini de \mathcal{P} .

Les définitions suivantes sont extraites de [Mil02]. Soit $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ la base standard de \mathbb{R}^3 . Le plan \mathcal{P} est maintenant plongé dans $\mathcal{P}' \subset \mathbb{R}^3$ où \mathcal{P}' est le plan contenant l'origine de \mathbb{R}^3 et ayant comme base $(\mathbf{e}'_x, \mathbf{e}'_y)$ où $\mathbf{e}'_x = (0, -1/\sqrt{2}, 1/\sqrt{2})$ et $\mathbf{e}'_y = (\sqrt{2}/3, -1/\sqrt{6}, -1/\sqrt{6})$. Observez que $\mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3$ est un vecteur normal² de \mathcal{P}' . Chaque point $p = (p_x, p_y) \in \mathbb{R}^2$ est projeté sur $p' \in \mathcal{P}'$ avec $p' = p_x \mathbf{e}'_x + p_y \mathbf{e}'_y$.

Considérons la relation de dominance sur \mathbb{R}^3 : $p \succcurlyeq q$ si et seulement si $p_i \geq q_i$ pour chaque $i \in \{1, 2, 3\}$. Notez que deux points de \mathcal{P}' sont incomparables. Le *filtre* généré par un nuage de points S de \mathcal{P} est l'ensemble

$$\langle S \rangle = \{ \alpha \in \mathbb{R}^3 : \alpha \succcurlyeq v \text{ pour un } v \in S \} .$$

La frontière \mathfrak{S}_S de $\langle S \rangle$ est la *surface orthogonale coplanaire* engendrée par S . Notez que dans [Mil02, FZ08], les auteurs considèrent les *surfaces orthogonales*, dans un cas plus général où les éléments de S sont deux à deux incomparables, pas nécessairement dans le même plan de vecteur normal $\mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3$. La Figure 3.3 montre un exemple de surface orthogonale coplanaire.

2. I.e., $\forall p' = (p'_1, p'_2, p'_3) \in \mathcal{P}'$, $p'_1 \mathbf{e}_1 + p'_2 \mathbf{e}_2 + p'_3 \mathbf{e}_3 = 0$.

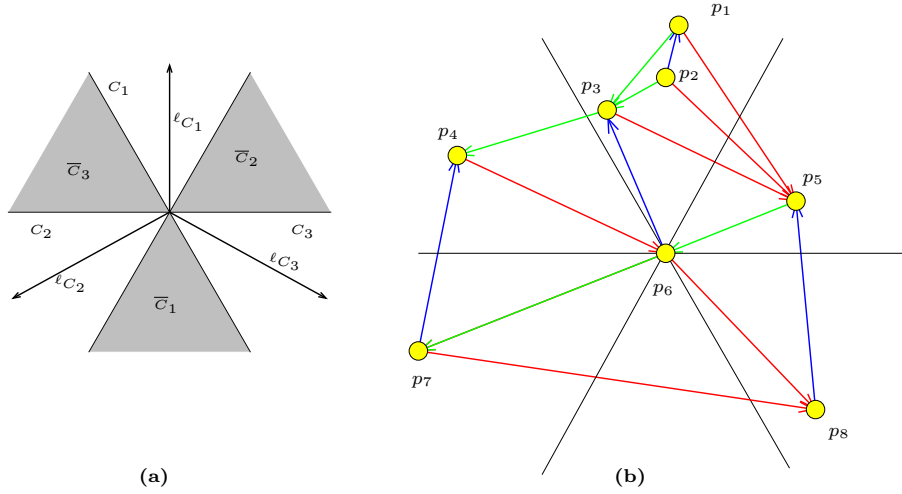


FIGURE 3.2 – (a) Illustration des notations pour les graphes demi- Θ_6 . (b) Un exemple de graphe demi- Θ_6 orienté.

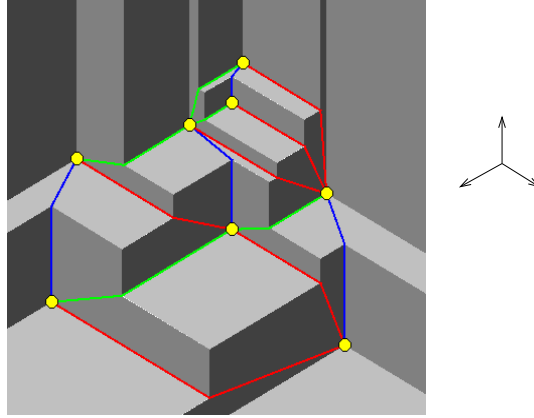


FIGURE 3.3 – Une surface orthogonale coplanaire avec son plongement géodésique.

Nous dénotons par $p \vee q$ le point $(\max\{p_1, q_1\}, \max\{p_2, q_2\}, \max\{p_3, q_3\})$. Si $p, q \in S$ et $p \vee q \in \mathfrak{S}_S$, alors \mathfrak{S}_S contient l'union des 2 segments de droites joignant p et q à $p \vee q$. Ces segments sont appelés *coude géodésique* de \mathfrak{S}_S .

L'*arc orthogonal* de $p \in S$ dans la direction du vecteur standard \mathbf{e}_i est le bout de rayon $p + \lambda \mathbf{e}_i$, $\lambda \geq 0$, qui suit un creux de \mathfrak{S}_S . Si $p \vee q$ est égal à $p + \lambda \mathbf{e}_i$, pour un certain $\lambda \geq 0$, on dit qu'il s'agit d'un coude de *type i* . Le coude géodésique correspondant est également de type i . Notez que $p \vee q$ partage 2 coordonnées (sur la base $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$) avec au moins 1 (et peut-être les 2) p et q . On dit qu'un coude géodésique est *simplement orienté* si son coude $p \vee q$ partage 2 de ses coordonnées avec p ou q pas les deux.

Une surface orthogonale \mathfrak{S}_S est *simplement orientée* si tous ces coudes géodésiques sont simplement orientés.

Théorème 3.1. [Fel01] ... Soit SW un bois de Schnyder. L'algorithme de comptage de face produit une surface géodésique SS . De plus les coudes géodésiques correspondent aux arêtes (orientation + coloration) de SW .

Définition 3.2. Soit S un nuage de points de \mathcal{P} tel que la surface orthogonale \mathfrak{S}_S est simplement orientée. Le plongement géodésique de S et les graphes orientés $\overrightarrow{\text{Geo}}(S)$ définis de la manière suivante :

- les sommets de $\overrightarrow{\text{Geo}}(S)$ sont les points de S et
- il y a un arc de p à q colorié i si et seulement si $p \vee q$ est un coude de type i

3.2.3 Triangulation de TD-Delaunay

Nous rappelons ici la définition de triangulation de TD-Delaunay introduite dans [Che89]. Il s'agit d'une variante des triangulations de Delaunay définie à partir d'une forme convexe [BCCS08].

Soit T (resp. \tilde{T}) le triangle équilatéral de côté 1, ayant comme centre de gravité l'origine et ayant 1 de ses sommets sur l'axe positif (resp. négatif) des y . Une *homothétie* de T est obtenue par un agrandissement de T suivie d'une translation : $p + \lambda T = \{p + \lambda z : z \in T\}$. La *distance triangulaire* entre 2 points p et q , notée $d_T(p, q)$, est définie de la manière suivante :

$$d_T(p, q) = \min \{ \lambda : \lambda \geq 0 \text{ and } q \in p + \lambda T \}$$

Remarque : la distance triangulaire n'est pas à proprement parler une distance car ce n'est pas une fonction symétrique. En général $d_T(p, q) \neq d_T(q, p)$.

Soit S un nuage de points de \mathcal{P} . Pour chaque point $p \in S$, $V_T(p)$ est la *cellule de TD-Voronoi* de p , définie par

$$V_T(p) = \{x \in \mathcal{P} : \text{pour chaque } q \in S, d_T(p, x) \leq d_T(q, x)\}.$$

Le *diagramme de TD-Voronoi* associé à S est la décomposition de \mathcal{P} en cellules de TD-Voronoi.

La Figure 3.4(a) montre un exemple de diagramme de TD-Voronoi. Observez que l'intersection de 2 cellules de TD-Voronoi n'est pas nécessairement de surface nulle. Par exemple si nous considérons $S = \{u = (-\sqrt{3}, 1), v = (\sqrt{3}, 1)\}$ (voir Figure 3.4(b)). L'intersection $V_T(u) \cap V_T(v)$ est égale à la partie du plan \mathcal{P} situé sous les lignes (o, u) et (o, v) où $o = (0, 0)$.

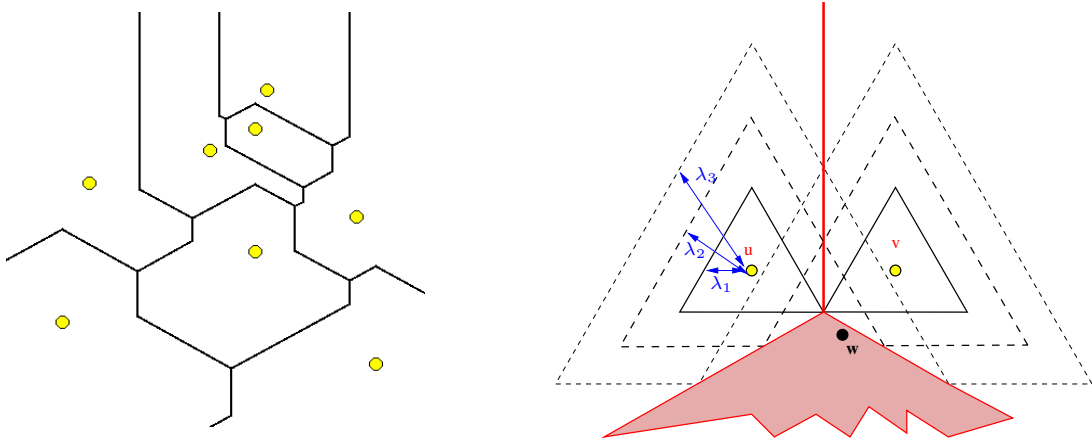


FIGURE 3.4 – (a) Diagramme de TD-Voronoi. (b) $\lambda_1 < \lambda_2 < \lambda_3$ représentent 3 distances triangulaires. L'ensemble $\{u, v\}$ est un ensemble de points générés. Le point w appartient aux cellules de TD-Voronoi de u et de v .

Comme nous considérons ici que des nuages de points en position générale, l'intersection de 2 cellules de TD-Voronoi est toujours de surface nulle.

Définition 3.3. Soit S un nuage de points de \mathcal{P} . La triangulation de TD-Delaunay de S , notée $\mathbf{TDDel}(S)$, est définie de la manière suivante :

- l'ensemble des sommets de $\mathbf{TDDel}(S)$ est S ; et
- (p, q) est une arête de $\mathbf{TDDel}(S)$ si et seulement si $V_T(p) \cap V_T(q) \neq \emptyset$.

3.3 Unification des concepts

Théorème 3.2. Soit S un nuage de points en position générale du plan \mathcal{P} . Soient $\mathbf{Geo}(S)$, resp. $\frac{1}{2}\Theta_6(S)$ les graphes non-orientés et non-coloriés sous-jacents de $\mathbf{Geo}(S)$, resp. $\frac{1}{2}\Theta_6(S)$.

$$\frac{1}{2}\Theta_6(S) = \mathbf{Geo}(S) = \mathbf{TDDel}(S) .$$

De plus,

$$\overrightarrow{\mathbf{Geo}}(S) = \frac{1}{2}\overrightarrow{\Theta_6}(S) .$$

Étant donnés deux points p et $q \in C_i^p$, on note $T_i(p, q)$ l'ensemble des points de \mathcal{P} dans $C_i^p \setminus \{p\}$ ayant un projeté orthogonal sur $\ell_{C_i^p}$ strictement plus proche de p que le projeté orthogonal de q sur $\ell_{C_i^p}$.

Lemme 3.1. Soit S un nuage de points en position générale du plan \mathcal{P} , $p \vee q$ est un coude de type i si et seulement si $q \in C_i^p$ et $T_i(p, q) \cap S = \emptyset$.

Preuve Soit S un nuage de points du plan \mathcal{P} et soient p et q deux points distincts de S . Sans perte de généralité, supposons que p est le point de coordonnées $(0, 0, 0)$ et que (q_1, q_2, q_3) sont les coordonnées de q . Notons que le cône C_p^i peut être décrit comme l'ensemble des points $r = (r_1, r_2, r_3)$ tels que $r_i \geq 0$ et $r_j \leq 0$ pour $j \neq i$. De plus le point p est le seul point r de C_p^i tel que $r_i = 0$, et que l'intérieur du cône C_p^i est exactement l'ensemble des points r tels que $r_i > 0$ et $r_j < 0$ pour $j \neq i$. Ces remarques sont illustrées par la Figure 3.5. Maintenant prouvons ce lemme pour $i = 1$ (les deux autres cas étant similaires).

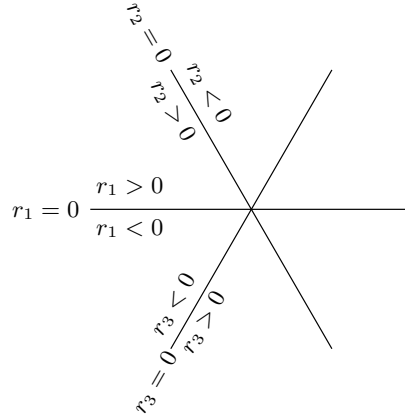


FIGURE 3.5 – Partition du plan \mathcal{P} en fonction du signe des coordonnées d'un point $r = (r_1, r_2, r_3)$ de \mathcal{P} .

Par définition, $p \vee q$ est un coude de type 1 si et seulement si

1. le point $p \vee q$ partage les 2 coordonnées d'indice 2 et 3 avec le point p et
2. $p \vee q$ est sur la surface orthogonale \mathfrak{S}_S .

Supposons maintenant que $q \in C_1^p$, c'est-à-dire que $p \vee q = (q_1, 0, 0)$. Le point $p \vee q$ est sur la surface orthogonale \mathfrak{S}_S si et seulement si il est sur la frontière du filtre généré par S , c'est-à-dire qu'il n'existe pas de point $r \in S$ tel que $r_1 < q_1$ et $r_2, r_3 < 0$. Comme l'ensemble des points r tels que $r_1 < q_1$ et $r_2, r_3 < 0$ est exactement $T_1(p, q)$, nous avons que $p \vee q$ est sur \mathfrak{S}_S si et seulement si $T_1(p, q) \cap S = \emptyset$. Ceci termine notre preuve. \square

Preuve du Théorème 3.2

L'égalité $\overrightarrow{\mathbf{Geo}}(S) = \frac{1}{2}\overrightarrow{\Theta_6}(S)$ découle du Lemme 3.1 et de l'observation suivante :

Il y a un arc (p, q) de couleur i dans $\frac{1}{2}\overrightarrow{\Theta_6}(S)$ si et seulement si $q \in C_i^p$ et $T_i(p, q) \cap S = \emptyset$.

Pour montrer que $\mathbf{Geo}(S) = \mathbf{TDDel}(S)$ il suffit de remarquer que :

- Chaque arête e de $\frac{1}{2}\Theta_6(S)$ est contenue dans un triangle vide T , tel que une des extrémités de e est un sommet de \bar{T} .
- Pour chaque arête e de $\mathbf{TDDel}(S)$ est contenue dans un triangle T , il existe un triangle T' inclus dans T tel qu'une des extrémité de e est un sommet de T' .

□

3.4 Applications

3.4.1 Spanner

Dans [Che89] Chew montre que les triangulations de TD-Delaunay sont des 2-spanners. En utilisant ce résultat, le Théorème 3.2 et en remarquant que le graphe Θ_6 est l'union de 2 demi- Θ_6 (l'un utilisant les cônes pairs et l'autre les cônes impairs) on obtient directement le corollaire suivant :

Corollaire 3.1. *Tout graphe demi- Θ_6 (et aussi tout graphe Θ_6) est un 2-spanner. De plus le graphe Θ_6 est l'union de graphes planaires.*

Observons que la borne de 2 sur l'étirement est la meilleure possible pour les graphes demi- Θ_6 . En effet, si on prenant 3 points formant quasiment un triangle équilatéral pointant vers le bas on peut obtenir un graphe demi- Θ_6 d'étirement arbitrairement proche de 2.

3.4.2 Réalisabilité de Delaunay

En utilisant l'algorithme par comptage de faces (cf. Section 2.2.2), Felsner et Zickfeld [FZ08, Theorem 10] ont montré que pour toute triangulation du plan G , il existe un nuage de points S tel que $\mathbf{Geo}(S) = G$. De plus ce nuage de points peut être calculé en temps linéaire. En utilisant l'équivalence entre plongement géodésique et triangulations de TD-Delaunay (Théorème 3.2) nous obtenons directement le résultat suivant :

Corollaire 3.2. *Toute triangulation du plan est TD-Delaunay réalisable.*

3.4.3 Algorithme de construction

Les algorithmes qui calculent les graphes Θ_k , les plongements géodésiques et les triangulations de TD-Delaunay ont été respectivement proposés dans [NS07, FZ08, CD85]. Ces algorithmes s'exécutent en $O(n \log n)$ et reposent essentiellement sur l'algorithme de "plane-sweep". L'unification des 3 concepts n'apporte donc rien de nouveau sur les algorithmes de construction de ces graphes.

3.5 Spanner de degré borné

Dans cette section nous nous intéressons à la question suivante (Problème ouvert numéro 14 du survey [BS])

Quel est le plus petit degré maximum qui puisse être atteint pour un spanner planaire ?

Comme nous l'avons évoqué dans l'introduction de ce chapitre la question a déjà fait l'objet de nombreuses contributions (voir Tableau 3.1). Toutes les contributions précédentes se basent sur les triangulations de Delaunay. Nous allons voir maintenant qu'en utilisant les triangulations de TD-Delaunay (et la richesse de leur structure), il est possible d'obtenir des 6-spanners planaires de degré maximum au plus 6 (la meilleure borne sur le degré qui prévalait était de 11).

Nous allons dans un premier temps présenter un algorithme simple permettant de calculer une 6-spanner de degré maximum au plus 9 (Section 3.6) puis nous allons voir comment descendre le degré maximum jusqu'à 6 (Section 3.7).

3.6 Un 6-Spanner simple planaire de degré maximum 9

Dans cette section nous montrons comment construire H_9 un sous-graphe de $\frac{1}{2}\overrightarrow{\Theta}_6$ de degré maximum au plus 9.

Dans cette construction, pour chaque noeud u , certains voisins de u jouent un rôle important. Pour $i \in \{1, 2, 3\}$, notons $children_i(u)$ l'ensemble des noeuds v tels que $u = parent_i(v)$. Notez que $children_i(u) \subseteq \overline{C}_i^u$. Dans $children_i(u)$, on distingue 3 sommets particuliers :

- $closest_i(u)$ le sommet de $children_i(u)$ ayant le plus proche projeté sur la bissectrice du cône \overline{C}_i^u ;
- $first_i(u)$ le premier sommet de $children_i(u)$ dans le sens anti-horaire en partant de l'axe x .
- $last_i(u)$ le dernier sommet de $children_i(u)$ dans le sens anti-horaire en partant de l'axe des x .

Notons que ces noeuds ne sont pas définis lorsque le cône \overline{C}_i^u est vide.

Soit (u, v) une arête telle que $v = parent_i(u)$. Un sommet t est i -pertinent par rapport à u si $t \in \overline{C}_i^v = \overline{C}_i^{parent_i(u)}$ ou $t = first_{i-1}(u) \neq closest_{i-1}(u)$ ou $t = last_{i+1}(u) \neq closest_{i+1}(u)$. Quand t est défini comme $first_{i-1}(u)$ ou $last_{i+1}(u)$, on omettra le "par rapport à u " ou on dira également que l'arête ut est i -pertinente. Par exemple, dans la Figure 3.6 (a), les sommets v_l et v_r sont i -pertinents par rapport à w . Dans la Figure 3.6 (b) le sommet $v_r = last_{i+1}(w)$ n'est pas i -pertinent puisque il n'est pas dans \overline{C}_i^w et $v_l = first_{i-1}(w)$ n'est pas i -pertinent puisqu'il est aussi $closest_{i-1}(w)$.

La construction de H_9 se décompose en 2 étapes qui peuvent facilement être effectuées de manière distribuée :

Etape 1. Calculer $\frac{1}{2}\overrightarrow{\Theta}_6$: pour chaque noeud $v \in S$, choisir $parent_i(u)$ dans chaque cône C_i^v .

Etape 2. Soit H_9 le graphe obtenu en choisissant des arêtes de $\frac{1}{2}\overrightarrow{\Theta}_6$ de la manière suivante : pour chaque sommet u et chaque cône négatif \overline{C}_i^u :

- ajouter l'arête $(u, closest_i(u))$ si $closest_i(u)$ existe,
- ajouter l'arête $(u, first_i(u))$ si $first_i(u)$ existe et si cette arête est $(i+1)$ -pertinent,
- ajouter l'arête $(u, last_i(u))$ si $last_i(u)$ existe et si cette arête est $(i-1)$ -pertinent.

Notez que qu'on peut tout de suite observer que H_9 est de degré maximum au plus 12 : chaque sommet a au plus 1 arête entrante par cône positif et 3 arêtes entrantes par cône négatif. Avec une analyse un peu plus fine du degré on montre que

Théorème 3.3. Le graphe H_9

- a degré maximum au plus 9,
- est un 3-spanner de $\frac{1}{2}\overrightarrow{\Theta}_6$, et est donc un 6-spanner.

La suite de cette section est dédiée à la preuve de ce résultat.

Charge d'un cône. Dans le but de borner le degré d'un noeud de H_9 , nous proposons un comptage des arcs d'incidents à un sommet. Chaque arête incidente à un noeud *charge* un des cônes de ce noeud de la manière suivante :

- chaque cône négatif \overline{C}_i^u est chargé par l'arête $(u, closest_i(u))$ si $closest_i(u)$ existe.
- chaque cône positif C_i^u est chargé par l'arête $(u, parent_i(u))$ si cette arête est dans H_9 , par l'arête $(u, first_{i-1}(u))$ si $first_{i-1}(u)$ est i -pertinent et par l'arête $(u, last_{i+1}(u))$ si $last_{i+1}(u)$ est i -pertinente.

Par exemple, dans la Figure 3.6 (a), le cône C_i^{vw} est chargé 2 fois : une fois par $v_l w$ et une fois par $v_r w$; le cône C_i^{vw} est chargé une fois par l'arête la plus courte. Dans (b), le cône C_i^{vw} n'est pas chargé du tout : l'arête $v_l w$ est la plus courte dans \overline{C}_{i-1}^w . Dans (c), le cône C_i^{vw} est chargé une fois par l'arête $v_l w$ et une fois par l'arête wu .

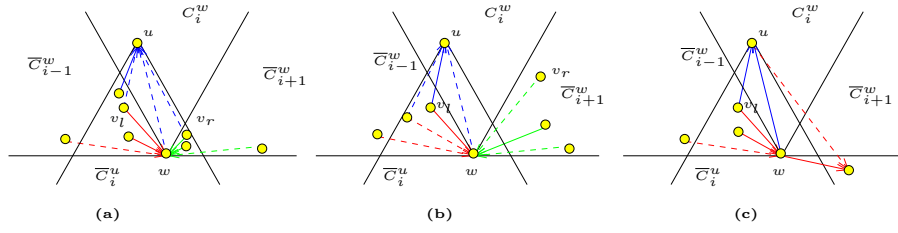


FIGURE 3.6 – Dans chacun des 3 cas, l'arête wu est dans $\frac{1}{2}\overrightarrow{\Theta}_6$ mais $w \neq \text{closest}_i(u)$. Les arêtes en trait plein sont les arêtes de H_9 . **(a)** Les arêtes ww_l et ww_r sont respectivement la première et la dernière (dans le sens anti-horaire) dans le cône C_{i+1}^w . De plus elles sont i -pertinentes par rapport à w . **(b)** L'arête ww_r n'est pas i -pertinente car ww_r n'est pas dans \overline{C}_i^u . L'arête ww_l est dans H_9 mais n'est pas i -pertinente car elle est la plus courte dans \overline{C}_{i-1}^w . **(c)** L'arête ww_l est i -pertinente. Notez que l'arête wu est dans H_9 car elle est $(i-1)$ -pertinente par rapport à u .

Nous notons $\text{charge}(C)$ la charge du cône C . Avec ce système de comptage, nous pouvons prouver le lemme suivant qui implique directement la première partie du Théorème 3.3 puisque la somme des charges des cônes autour d'un sommet est égale au degré de ce sommet dans H_9 .

Lemme 3.2. *Chaque cône négatif est chargé par au plus 1 arête et chaque cône positif est chargé par au plus 2 arêtes.*

Preuve La charge d'un cône négatif est par construction clairement d'au plus 1. Pour prouver ce lemme, il nous faut simplement montrer qu'un cône positif ne peut pas être chargé 3 fois. Soit C_i^w un cône positif.

Soit $u = \text{parent}_i(w)$. Si l'arête (w, u) n'est pas dans H_9 alors, clairement $\text{charge}(C_i^w) \leq 2$. Dans le cas contraire, considérons les 3 cas suivants :

Cas 1 : $w = \text{closest}_i(u)$. Tout point de $R = \overline{C}_i^u \cap \{\overline{C}_{i-1}^w \cup \overline{C}_{i+1}^w\}$ est plus proche de u que de w . Comme w est le plus proche voisin de u dans le cône \overline{C}_i^u , les noeuds $\text{first}_{i-1}(w)$ et $\text{last}_{i+1}(w)$ ne peuvent pas être i -pertinents. Donc les sommets $\text{first}_{i-1}(w)$ et $\text{last}_{i+1}(w)$ ne sont pas i -pertinents. Donc $\text{charge}(C_i^w) = 1$.

Cas 2 : $w = \text{last}_i(u)$ et w est $(i-1)$ -pertinent par rapport à u (voir Figure 3.6 (c)). Dans ce cas, w, u et $\text{parent}_{i-1}(w) = \text{parent}_{i-1}(u)$ forment un triangle vide dans $\frac{1}{2}\overrightarrow{\Theta}_6$. Par conséquent, $\overline{C}_i^u \cap \overline{C}_{i+1}^w$ est vide. Donc $\text{last}_{i+1}(w)$ n'est pas i -pertinent. Donc $\text{charge}(C_i^w) \leq 2$.

Cas 3 : $w = \text{first}_i(u)$ et w est $(i+1)$ -pertinent. Par symétrie, ce cas se ramène au cas 2.

□

La preuve ci-dessus donne une information structurelle supplémentaire qui nous sera utile dans la section suivante :

Corollaire 3.3. *Soit $u = \text{parent}_i(w)$. Si $\text{charge}(C_i^w) = 2$ alors soit :*

1. (w, u) n'est pas dans H_9 , et $\text{first}_{i-1}(w)$ et $\text{last}_{i+1}(w)$ sont i -pertinent (et sont donc des voisins de w dans H_9), ou bien
2. $w = \text{last}_i(u)$ est $(i-1)$ -pertinent et $\text{first}_{i-1}(w)$ est i -pertinent (et donc (w, u) et $(\text{first}_{i-1}(w), w)$ sont dans H_9), ou bien
3. $w = \text{first}_i(u)$ est $(i+1)$ -pertinent et $\text{last}_{i+1}(w)$ est i -pertinent (et donc (w, u) et $(\text{last}_{i+1}(w), w)$ sont dans H_9).

Dans le cas 1, notons que les sommets $\text{first}_{i-1}(w)$, w , et $\text{last}_{i+1}(w)$ sont dans \overline{C}_i^u et que u est plus proche de $\text{first}_{i-1}(w)$ et de $\text{last}_{i+1}(w)$ que de w . Quand le cas 1 se produit, on dit que w est i -distant.

Pour montrer que H_9 est un 3-spanner de $\frac{1}{2}\overrightarrow{\Theta}_6$, nous allons montrer que pour chaque arête wu qui est dans $\frac{1}{2}\overrightarrow{\Theta}_6$ mais pas dans H_9 , il y a un chemin dans H_9 de u à w de longueur au plus $3|uw|$. Soit wu

une arête entrante de u dans $\frac{1}{2}\overrightarrow{\Theta}_6$. Comme $wu \notin H_9$, l'arête entrante la plus courte de $\frac{1}{2}\overrightarrow{\Theta}_6$ dans le cône C de u doit être dans H_9 : appelons cette arête vu . Sans perte de généralité, nous supposons que vu est avant wu dans cône C le sens anti-horaire autour de u .

Considérons toutes les arêtes de $\frac{1}{2}\overrightarrow{\Theta}_6$ incidentes à u qui sont dans le cône \overline{C}_i^u entre les arêtes vu et wu : $vu = v_0u, v_1u, \dots, v_ku = wu$ (toujours dans le sens anti-horaire). Comme $\frac{1}{2}\overrightarrow{\Theta}_6$ est une triangulation, le chemin $v_0v_1, v_1v_2, \dots, v_{k-1}v_k$ est dans $\frac{1}{2}\overrightarrow{\Theta}_6$. Nous appelons ce chemin, le *chemin canonique* de w à u (voir Figure 3.7).

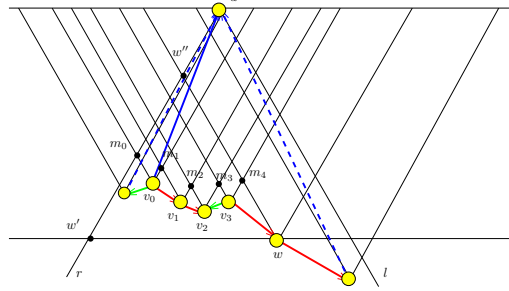


FIGURE 3.7 – Chemin Canonique.

Lemme 3.3. Soit $(w, u = \text{parent}_i(w))$ une arête de $\frac{1}{2}\overrightarrow{\Theta}_6$ et $v = \text{closest}_i(u)$. Si $w \neq v$ alors :

1. H_9 contient l'arête vu et le chemin canonique de w à u ,
2. $|uv| + \sum_{i=1}^k |v_{i-1}v_i| \leq 3|uw|$.

La seconde partie du Théorème 3.3 se déduit directement du Lemme 3.3.

Preuve du Lemme 3.3

Soit $e = (v_j, v_{j+1})$ une arête du chemin canonique allant de w à u . Supposons dans un premier temps que e est entrante en v_j . Observons que v_{j+1} est le voisin de v_j juste avant u dans l'ordre anti-horaire autour de v_j dans la triangulation $\frac{1}{2}\overrightarrow{\Theta}_6$. Donc $v_{j+1} = \text{last}_{i+1}(v_j)$. Comme v_{j+1} est dans \overline{C}_i^u , v_{j+1} est i -pertinent (par rapport à v_j) ou bien $v_{j+1} = \text{closest}_{i+1}(v_j)$. Dans les 2 cas, e est dans H_9 . Maintenant supposons que e est entrante en v_{j+1} . Nous montrons de manière similaire que $v_j = \text{first}_{i-1}(v_{j+1})$ et que v_j est i -pertinent (par rapport à v_{j+1}) ou bien que $v_j = \text{closest}_{i-1}(v_{j+1})$. Dans les 2 cas e est dans H_9 . Ceci prouve la première partie du lemme.

Pour montrer la deuxième partie du lemme, nous notons $C_i^{v_j}$ le cône contenant u du sommet v_j du chemin canonique, pour $j = 0, 1, \dots, k$. Notons r_j et l_j les rayons définissant les frontières du cône $\overline{C}_i^{v_j}$. Soit r et l les rayons définissant les frontières du cône \overline{C}_i^u . Soit m_0 le point d'intersection du rayon r avec l_0 et soit m_j le point d'intersection des rayons r_{j-1} et l_j pour chaque $1 \leq j \leq k$. Soit w' l'intersection du rayon r avec la ligne perpendiculaire à l_C ($C = C_i^w$) passant par w et soit w'' le point d'intersection de l_k avec r (voir Figure 3.7).

Remarquons que $|uv| = |uv_0| \leq |um_0| + |m_0v_0|$, et

$$|v_{j-1}v_j| \leq |v_{j-1}m_j| + |m_jv_j|$$

pour tout $1 \leq j \leq k$. De plus $|uv_0| \geq |um_0|$. Donc

$$\begin{aligned} |uv_0| + \sum_{j=1}^k |v_{j-1}v_j| &\leq |um_0| + \sum_{j=0}^k |m_jv_j| + \sum_{j=0}^{k-1} |v_jm_{j+1}| \\ &\leq |um_0| + |ww''| + |w''m_0| \\ &\leq |uw'| + |ww'| + |w''w'| \\ &\leq |uw'| + 2|ww'|. \end{aligned}$$

Observons que $|uw| = \sqrt{(|uw'| \cos \pi/6)^2 + (|ww'| - |ww'|/2)^2}$. Soit $\alpha = |ww'|/|uw'|$; remarquons que $0 \leq \alpha \leq 1$. On en déduit que

$$\begin{aligned} \frac{|uw'| + 2|ww'|}{|uw|} &\leq \frac{(1 + 2\alpha)|uw'|}{\sqrt{(|uw'| \cos \pi/6)^2 + ((\alpha - 1/2)|uw'|)^2}} \\ &\leq \frac{1 + 2\alpha}{\sqrt{1 - \alpha + \alpha^2}} \\ &\leq \max_{\alpha \in [0..1]} \left\{ \frac{1 + 2\alpha}{\sqrt{1 - \alpha + \alpha^2}} \right\} \\ &\leq 3. \end{aligned}$$

□

3.7 Un 6-Spanner planaire de degré maximum 6

Maintenant que nous avons montré que H_9 était un 6-spanner de degré au plus 9, nous allons voir comment en supprimant et en ajoutant quelques arêtes à ce graphe, on peut construire un 6-spanner de degré maximum au plus 6. Nous effectuons cela en essayant de diminuer le nombre d'arêtes qui chargent un cône positif jusqu'à 1. Nous ne pouvons pas faire cela directement, mais nous pouvons amortir les cônes positifs chargés 2 à l'aide de cônes négatifs chargés 0. Grâce au Corollaire 3.3, nous avons simplement à nous préoccuper des deux premiers cas (le 3ième étant symétrique au second).

Avant de présenter la construction finale, commençons par une propriété structurelle des cônes positifs de H_9 chargés 2. Pour mémoire, on dit qu'un sommet est i -distant s'il a 2 voisins i -pertinents dans H_9 (ce qui correspond au cas 1 du Corollaire 3.3). Par exemple, dans la Figure 3.7, le sommet v_2 est i -distant.

Lemme 3.4 (Séquence de charge interdite). *Si, dans H_9 , $charge(C_i^w) = 2$ et w n'est pas un sommet i -distant :*

- $first_{i-1}(w)$ est i -pertinent, $charge(C_{i-1}^w) \leq 1$ et $charge(\overline{C}_{i+1}^w) = 0$ ou
- $last_{i+1}(w)$ est i -pertinent, $charge(C_{i+1}^w) \leq 1$ et $charge(\overline{C}_{i-1}^w) = 0$.

Preuve Grâce au Corollaire 3.3, si w n'est pas un sommet i -distant, alors $first_{i-1}(w)$ ou $last_{i+1}(w)$ est i -pertinent. Nous supposons le second cas (le premier suivra par symétrie).

Nous commençons par prouver l'existence d'un cône de charge 0. Si $u = parent_i(w)$, alors grâce au Corollaire 3.3, $w = last_i(u)$ et w est $(i-1)$ -pertinent (par rapport à u). Cela signifie que les sommets w , u , et $v = parent_{i-1}(u) = parent_{i-1}(w)$ forment un triangle vide dans $\frac{1}{2}\overrightarrow{\Theta}_6$ et donc il n'y pas d'arête qui arrive dans \overline{C}_{i+1}^w . Par conséquent $charge(\overline{C}_{i+1}^w) = 0$.

Maintenant prouvons par l'absurde que $charge(C_{i-1}^w) \leq 1$. Supposons que $charge(C_{i-1}^w) = 2$. Grâce au Corollaire 3.3 il y a 3 cas à considérer. Nous venons de prouver qu'il n'y a pas d'arête dans \overline{C}_{i+1}^w , et donc il ne peut pas y avoir de sommet $first_{i+1}(w)$ et les 2 premiers cas ne peuvent pas se produire. Le cas 3 du Corollaire 3.3 implique que $w = first_{i-1}(v)$, ce qui n'est pas possible car l'arête (u, v) est avant

(w, u) dans le sens anti-horaire dans $\overline{C}_{i-1}(v)$. \square

Etape 3. On construit H_{S_3} à partir H_9 de la manière suivante : pour chaque entier $1 \leq i \leq 3$ et pour chaque sommet i -distant w :

- ajouter l'arête $(\text{first}_{i-1}(w), \text{last}_{i+1}(w))$ à H_{S_3} ;
- soit w' le sommet de $\{\text{first}_{i-1}(w), \text{last}_{i+1}(w)\}$ qui est à l'indice le plus grand dans le chemin canonique. Supprimer l'arête (w, w') de H_{S_3} .

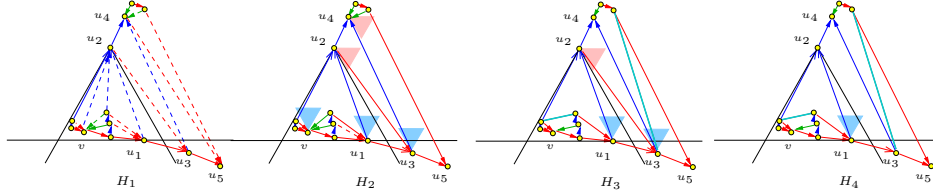


FIGURE 3.8 – De $\frac{1}{2}\overrightarrow{\Theta}_6$ (les flèches pleines sont les plus proches arêtes) à H_9 . Les cônes positifs bleus ciel et roses ont une charge de 2. Le sommet v est i -distant et le sommet u_4 est $i+1$ -distant.

Nouvel assignement des charges. Comme une nouvelle arête e est ajoutée entre $\text{first}_{i-1}(w)$ et $\text{last}_{i+1}(w)$ à l'étape 3, nous assignons la charge de e à $\overline{C}_{i+1}^{\text{first}_{i-1}(w)}$ et à $\overline{C}_{i-1}^{\text{last}_{i+1}(w)}$. Nous notons $\widetilde{\text{charge}}(C)$ la nouvelle charge de C dans H_{S_3} ainsi que dans le graphe dans le nouveau graphe H_6 que nous allons construire. Le lemme suivant montre que l'application de l'étape 3 ne crée pas de charge 2 et fait passer la charge du cône C_i^w d'un sommet i -distant s de 2 à 1.

Lemme 3.5 (Sommet distant). *Si w est un sommet i -distant alors :*

- $\widetilde{\text{charge}}(C_i^w) = \text{charge}(C_i^w) - 1 = 1$;
- $\widetilde{\text{charge}}(\overline{C}_{i+1}^{\text{first}_{i-1}(w)}) = \text{charge}(\overline{C}_{i-1}^{\text{last}_{i+1}(w)}) = 1$.

Etape 4. Nous construisons H_6 à partir de H_{S_3} de la manière suivante : pour chaque entier $1 \leq i \leq 3$ et chaque sommet w tel que $\widetilde{\text{charge}}(C_i^w) = 2$ et $\text{charge}(\overline{C}_{i-1}^w) = \text{charge}(\overline{C}_{i+1}^w) = 1$, si $w = \text{last}_i(\text{parent}_i(w))$ alors supprimer l'arête $(w, \text{first}_{i-1}(w))$ de H_{S_3} sinon supprimer $(w, \text{last}_{i+1}(w))$.

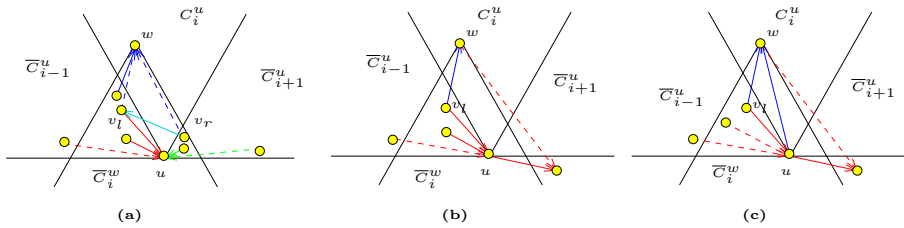


FIGURE 3.9 – (a) L'étape 3 est appliquée à la configuration de la Figure 3.6 (a) : l'arête wv_r est supprimée car le chemin canonique de w à u ne l'utilise pas. L'arête est alors remplacée par l'arête $v_r v_l$. (b) L'étape 4 appliquée à la Figure 3.6 (c) : l'arête wv_l est supprimée. (c) l'arête $v_l w$ est la plus courte dans \overline{C}_{i-1}^w ; le cône C_i^w est donc chargé 1 par wu , et l'arête $v_l w$ n'est pas supprimée à l'étape 4.

Lemme 3.6. *Il y a une injection entre les cônes positifs C_i^w chargés 2 à l'étape 4 et les cônes négatifs de w chargés 0.*

Preuve Le Corollaire 3.3 donne 2 types de propriétés pour les cônes chargés 2 dans H_6 . Si le cône C_i^w est l'un des cônes d'un sommet i -distance w de H_6 alors C_i^w aura une charge de 1 après l'étape 3

d'après le Lemme 3.5. Si w est i -distant, il peut y avoir 2 cas d'après le Lemme 3.4. Nous supposons que nous sommes dans le premier cas (le deuxième s'obtenant directement par symétrie); Supposons donc que $\overrightarrow{C}_{i+1}^w$ a une charge de 0 dans H_9 . Si la charge de ce cône passe à 1 à l'étape 3 alors l'étape 4 fera décroître la charge de C_i^w à 1. Donc, si C_i^w a toujours une charge de 2 à l'étape 4, alors $\overrightarrow{C}_{i+1}^w$ aura toujours une charge de 0 et nous ferons correspondre C_i^w avec ce cône négatif. Le seul autre cône positif pouvant être associé à $\overrightarrow{C}_{i+1}^w$ est le cône C_{i-1}^w , mais ce cône a une charge d'au plus 1 d'après le Lemme 3.4. \square

Théorème 3.4. H_6 est un 6-spanner planaire de \mathcal{E} de degré maximum 6.

Preuve D'après le Corollaire 3.3, le Lemme 3.5, et surtout d'après le Lemme 3.6, le degré maximum de H_6 est d'au plus 6.

Montrons maintenant que H_6 est un 6-spanner. D'après le Lemme 3.3, pour chaque arête de $\frac{1}{2}\overrightarrow{\Theta}_6$ qui n'est pas dans H_9 , le chemin canonique allant de w à u dans H_9 a une longueur d'au plus $3|wu|$. Nous allons montrer que la suppression d'arêtes à l'étape 3 est compensée par l'ajout d'autres arêtes également à l'étape 3. Observons d'abord qu'à chaque fois qu'une arête de H_9 est supprimée à l'étape 3, à chaque fois un raccourci est ajouté. De plus à l'étape 4 d'autres arêtes sont également supprimées. Ces arêtes supprimées sont toujours les premières arêtes d'un chemin canonique de w à u , ou l'arête wu est dans H_9 et en plus elle est soit la première, soit la dernière d'un cône négatif de u . Cela signifie que l'arête du chemin canonique n'est utile (dans nos preuves) uniquement pour relier w à u et à aucune autre paire de sommets. Donc cette arête peut être supprimée à l'étape 4 car l'arête $wu \in H_9$. En résumé, aucune arête "intermédiaire" du chemin canonique n'est supprimée sans qu'un raccourci ne soit ajouté, et que les arêtes d'un chemin canonique "final" ne seront supprimées que si elle ne sont plus utiles. Donc chaque chemin canonique (de longueur au plus $3|wu|$) de H_9 est remplacé par un chemin canonique de longueur au plus $3|wu|$. D'après le Lemme 3.3, l'argument similaire peut-être également utilisé pour chaque arête de H_9 qui n'est plus dans H_6 .

Il reste à montrer que H_6 est planaire. Plus précisément, nous devons montrer que les arêtes introduites à l'étape 3 ne créent pas de croisements dans H_{S3} . Soit $v_l v_r$ une arête ajoutée à l'étape 3. Observons que dans $\frac{1}{2}\overrightarrow{\Theta}_6$ il y a deux faces triangulaires adjacentes $f_1 = wv_l w$ et $f_2 = uvv_r$. Puisque l'arête wv_l est dans C_{i-1}^w et que v_r est dans C_{i+1}^w l'angle $v_l w v_r$ est plus grand que π . Donc l'arête $v_l v_r$ est dans les deux faces f_1 et f_2 . La seule arête de $\frac{1}{2}\overrightarrow{\Theta}_6$ qui coupe cette arête est l'arête wu . Comme l'arête wu n'est plus présente dans H_6 , il n'y a pas de croisement entre une arête de $\frac{1}{2}\overrightarrow{\Theta}_6 \cap H_{S3}$ et une arête ajoutée à l'étape 3. Il reste à montrer que 2 arêtes ajoutées à l'étape 3 ne se coupent pas. Soit $v'_l v'_r$ une arête créée à l'étape 3 et soient $f'_1 = u'v'_l w'$ et $f'_2 = u'w'v'_r$ les deux faces incidentes à cette arête dans $\frac{1}{2}\overrightarrow{\Theta}_6$. Si les arêtes $v_l v_r$ et $v'_l v'_r$ se croisent alors elles sont incidentes à au moins une même face de $\frac{1}{2}\overrightarrow{\Theta}_6$, i.e. $\{f_1, f_2\} \cap \{f'_1, f'_2\} \neq \emptyset$. Observons que les arêtes $v_l u, wu$ et $v_r u$ sont coloriées i , l'arête wv_l est coloriée $i - 1$ et l'arête wv_r est coloriée $i + 1$. De manière similaire, les arêtes $v'_l u', w'u'$ et $v'_r u'$ sont coloriées i' , l'arête $w'v'_l$ est coloriée $i' - 1$ et l'arête $w'v'_r$ est coloriée $i' + 1$. Chaque face f_1, f_2, f'_1 et f'_2 ont 2 arêtes de la même couleur, donc $i = i'$. A cause de la couleur de la troisième arête de chaque face, on a que $f_1 = f'_1$ et $f_2 = f'_2$, et donc $v_l v_r = v'_l v'_r$. Ceci montre bien que H_6 est planaire. \square

3.8 Conclusion et perspectives

Nous avons implémenté les algorithmes de construction des spanners de degré borné. Le programme est disponible à l'adresse suivante : <http://www.labri.fr/perso/bonichon/Bounded>.

Comme nous l'avons vu les demi- Θ_6 sont des graphes planaires. Les graphes de Yao [Yao82] sont très proches des Θ -graphes : dans chaque cône autour de p , p sélectionne le plus proche sommet de ce cône au lieu de prendre celui ayant le plus proche projeté sur ℓ_C comme c'est le cas pour les Θ -graphes. Les

demi- Y_6 peuvent être définis comme nous l'avons fait pour les demi- Θ -graphes en en considérant qu'un cône sur deux. Il est naturel de se poser la question de la planarité de ces graphes-ci. Malheureusement, les demi- Y_6 -graphes n'ont pas d'aussi belles propriétés structurelles et les demi- Y_6 -graphes ne sont pas planaires en général (voir Figure 3.10).

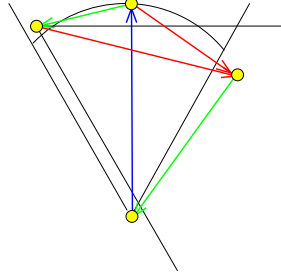


FIGURE 3.10 – Exemple de demi- Y_6 graphe qui n'est pas un graphe plan (mais il est planaire).

Γ -Delaunay réalisabilité Dans ce chapitre nous avons montré que toute triangulation du plan est TD -Delaunay réalisable. On peut naturellement se demander s'il existe une autre fonction de distance Γ telle que toute triangulation du plan est Γ -Delaunay réalisable? Sinon, existe-t-il une caractérisation simple des triangulations qui sont Γ -Delaunay réalisables?

Remarquons que K_4 (et plus généralement toute triangulation ayant un triangle séparateur) n'est pas L_1 -Delaunay réalisable (cf. Figure 3.11) alors qu'il est L_2 -Delaunay réalisable.

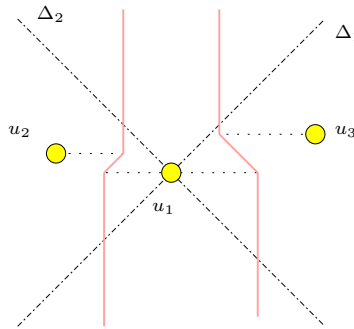


FIGURE 3.11 – Configuration impossible pour K_4 .

A l'inverse il existe des triangulations qui sont L_1 -Delaunay réalisables mais qui ne sont pas L_2 -Delaunay réalisables (voir Figure 3.12). Dans le cas de la L_1 -Delaunay réalisabilité, une première question à se poser est : est-ce que toutes les triangulations sans triangle séparateur sont L_1 -Delaunay réalisables? Un angle d'attaque possible est celui que nous avons utilisé sur les TD -Delaunay : ajouter de la structure aux triangulations. Dans les cas des triangulations non contraintes, nous avons utilisé les bois de Schnyder. Dans le cas des triangulations sans triangle séparateur, les structures adaptées semblent être les étiquetages réguliers d'arête [KH97] (*regular edge labeling*) ou de manière équivalente les *structures transverses* [Fus06]. Étant donnée une triangulation du plan sans triangle séparateur et ayant 4 sommets sur la face externe, une structure transverse est une orientation et une coloration des arêtes telle qu'autour de chaque sommet interne, on trouve une séquence non vide d'arêtes entrantes bleues, suivie d'une séquence non vide d'arêtes entrantes rouges, suivie d'une séquence d'arêtes sortantes bleues, suivie d'une séquence non vide d'arête sortantes rouges (voir Figure 3.13).

Comme nous l'avons déjà évoqué en conclusion des chapitres précédents, Gonçalves et Lévêque [GL12] ont proposé une élégante généralisation des bois de Schnyder aux cartes 3-connexes de genre 1. Les définitions des TD -Delaunay et des demi- Θ_6 -graphe se généralisent tout naturellement au cas du tore

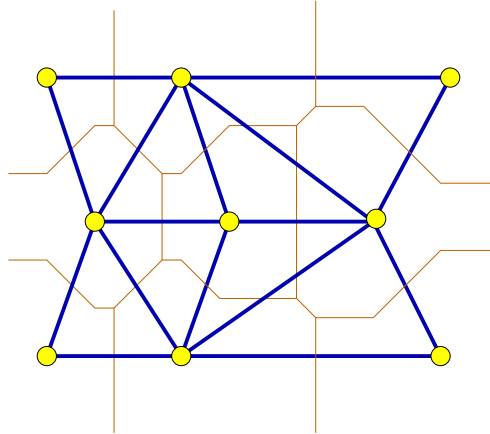


FIGURE 3.12 – Une triangulation de L_1 -Delaunay réalisable mais pas L_2 -Delaunay réalisable.

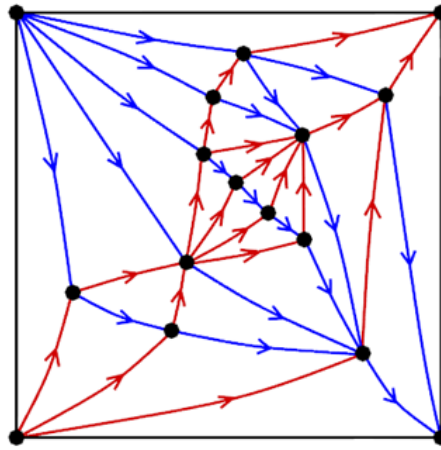


FIGURE 3.13 – Exemple de structure transverse (source : [Fus06]).

carré (le bois de Schnyder de la Figure 2.19 est bien un demi- Θ_6 graphe du tore carré). La question qui reste ouverte est de savoir si toutes les triangulations du tore sont TD-Delaunay réalisables.

A l'instar des triangulations de TD-Delaunay, le lien avec les structures transverses passe par des graphes "à la Yao" : les Yao_4^∞ . Le graphe Yao_4^∞ est défini de la manière suivante : chaque sommet p partitionne le plan en 4 cônes dont les bords sont parallèles aux axes et choisit dans chaque cône le sommet le plus proche selon la norme L_∞ (cf. Figure 3.14). Trivialement, ces graphes sont des sous-graphes des triangulations de L_∞ -Delaunay, car chaque arête est contenue dans un carré vide dont une des extrémités est un coin du carré. On peut utiliser ces cônes pour colorier et orienter les arêtes pour obtenir une structure transverse : les arêtes dans l'axe SO-NE sont coloriées en rouge et orientées vers le NE et les arêtes dans l'axe NO-SE sont coloriées en bleu et orientées vers le SE.

Degré borné Dans ce chapitre, nous avons proposé un algorithme pour calculer un 6-spanner planaire de degré au plus 6. L'analyse de l'étirement de notre construction est fine puisqu'il existe des nuages de points dont l'étirement du spanner construit peut-être arbitrairement proche de 6 (voir Figure 3.15).

Notre construction peut être utilisée pour obtenir un 6-spanner planaire de degré au plus 6 d'un graphe d'hexagones unitaires (Unit Hexagonal Graph), l'équivalent des graphes de disques unitaires où on a remplacé les disques par des hexagones (cf. Figure 3.16). Pour cela il suffit d'observer que si une

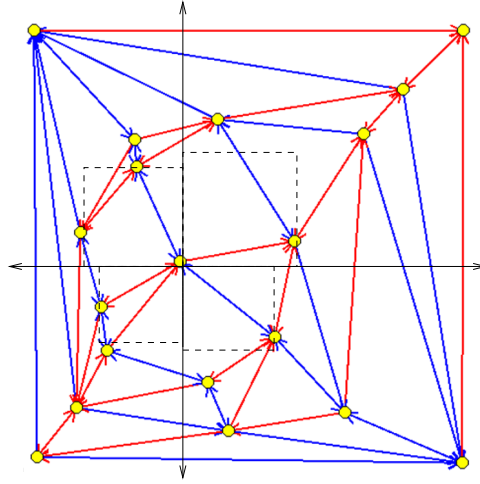


FIGURE 3.14 – Exemple de graphe Yao_4^∞ : chaque sommet choisit dans chaque quadrant le sommet le plus proche pour la norme L_∞ . Remarque : des sommets peuvent se choisir mutuellement.

arête e est dans $UHG \setminus H_6$ (où UHG désigne le graphe d’hexagones unitaires) alors, les arêtes du chemin canonique ont une “longueur” (longueur au sens de la distance hexagonale) plus petite que la “longueur” de l’arête supprimée.

Étirement vs. étirement routable Comme nous l’avons vu dans ce chapitre, les triangulations de TD-Delaunay sont des 2-spanners et ce facteur est le meilleur possible. Grâce à notre “bijection” avec les demi- Θ_6 graphes, nous avons pu étendre ce résultat au Θ_6 -graphes. Récemment, Damian et Raudonis [DR12] ont montré que les graphes Yao_6 étaient eux aussi des spanners montrant que l’étirement des graphes Yao_6 était une constante fois l’étirement des graphes Θ_6 .

Avec Ljubomir Perković, Cyril Gavoille et Nicolas Hanusse [BGHP12], nous nous sommes récemment intéressés aux triangulations de L_1 -Delaunay. Nous avons montré que les triangulations de L_1 -Delaunay étaient des $(\sqrt{4 + 2\sqrt{2}})$ -spanners ($\sqrt{4 + 2\sqrt{2}} \approx 2.61$) et que ce facteur est le meilleur possible, améliorant ainsi la borne initiale de Chew [Che86] de $\sqrt{10} \approx 3.16$.

Le fait d’avoir obtenu des bornes fines pour les triangulations de Delaunay définies à partir de triangles et de carrés vides nous incite à penser qu’une voie possible pour calculer l’étirement des triangulations de Delaunay serait de considérer les triangulations de Delaunay définies à partir d’un polygone régulier.

Récemment Bose et al. [BFvRV12] ont analysé l’étirement routable d’un graphe, c’est à dire l’étirement d’une route calculée par un algorithme de routage local (i.e. ne connaissant que les coordonnées de la source, de la destination, du sommet courant ainsi que ses voisins). Ils ont montré que bien que l’étirement des triangulations de TD-Delaunay soit 2, il n’existe pas d’algorithme local ayant un étirement meilleur que $5/\sqrt{3}$. De plus ils ont montré qu’il existait un algorithme local atteignant cette borne.

Ce paramètre d’étirement routable est encore peu étudié et nous semble être un axe de recherche assez vaste et tout à fait pertinent dans le contexte de l’algorithmique distribuée.

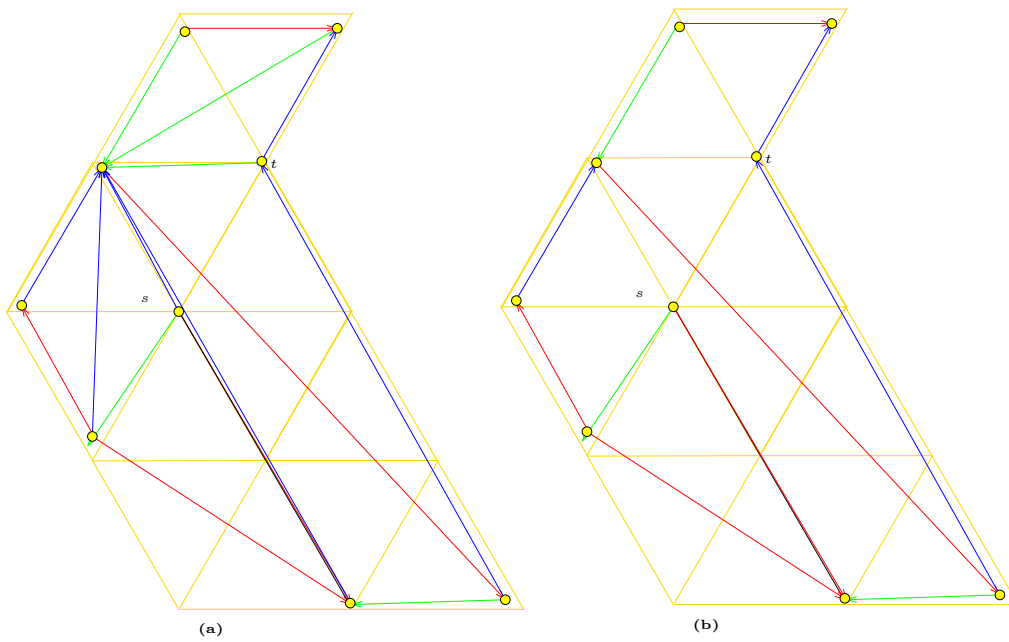


FIGURE 3.15 – (a) Une triangulation de TD-Delaunay d'étirement presque 2. (b) Le graphe H_6 correspondant avec un étirement de presque 6.

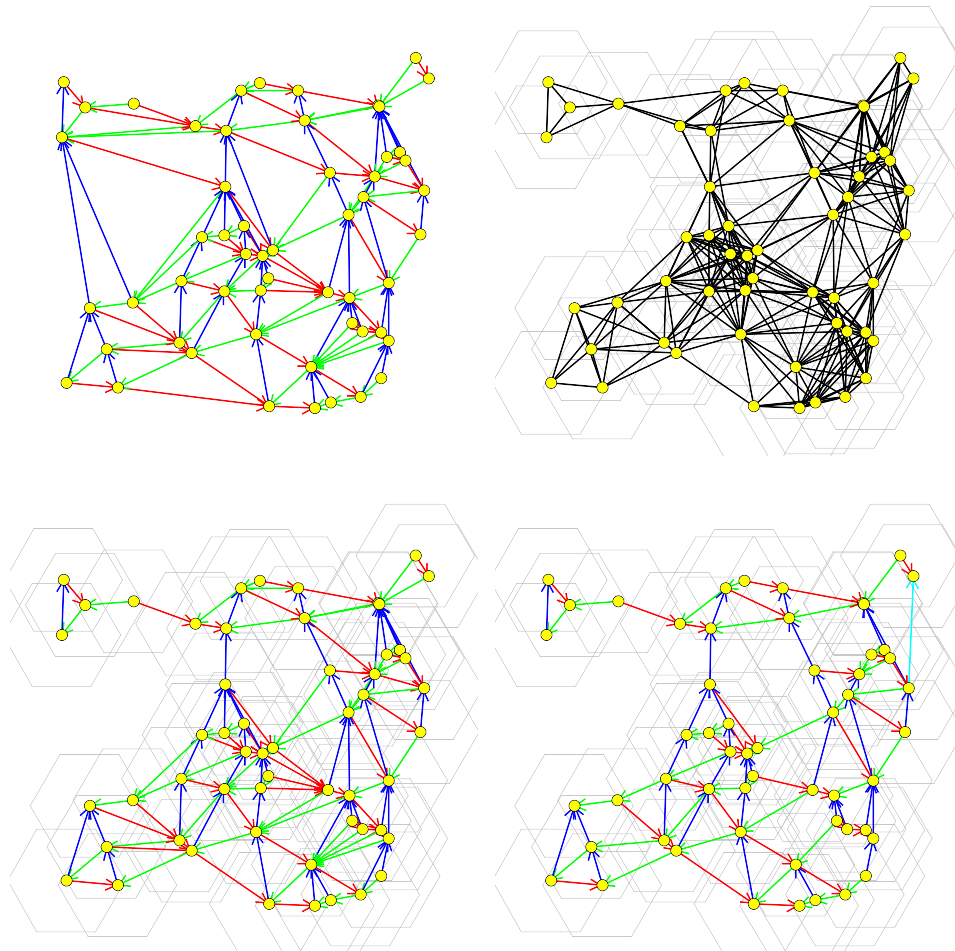


FIGURE 3.16 – (a) Une triangulation de TD-Delaunay. (b) Un UHG U . (c) $U \cap \mathbf{TDDel}$. (d) $U \cap H_6$.

Chapitre 4

Algorithmes d'approximation : Agrégation de ressources hétérogènes

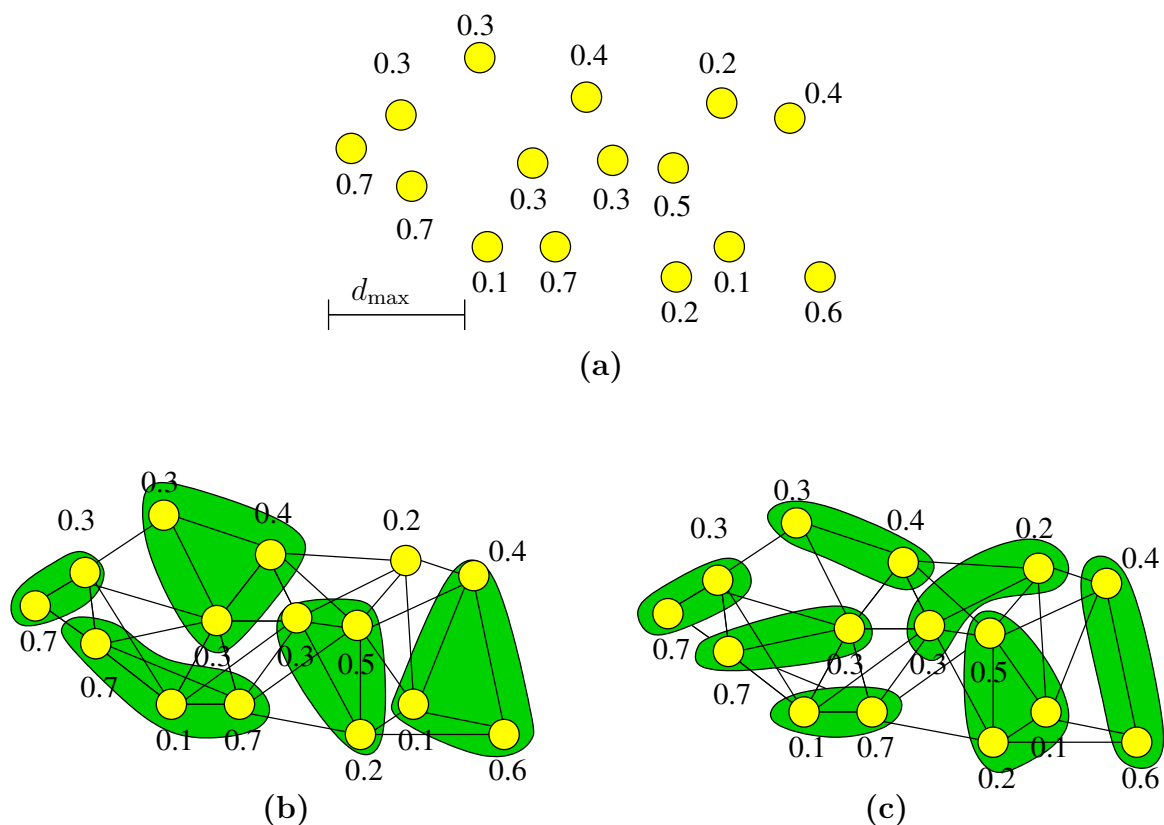


FIGURE 4.1 – (a) Un nuage de points pondéré avec une distance seuil d_{\max} . Le graphe représenté en (b) et (c) est le graphe de compatibilité (il y a une arête entre chaque paire de sommets à distance au plus d_{\max}). (b) Une solution (optimale) au problème de BCCD : chaque groupe (bin) forme une clique de poids au moins 1. (c) Une solution (optimale) au problème de BPCD : les sommets sont partitionnés en cliques de poids au plus 1.

Dans ce chapitre, les nuages de points que nous allons considérer sont formés de machines connectées à Internet. La distance entre ces points est la latence qui sépare ces machines : c'est à dire le temps que met un message élémentaire pour aller d'une machine à une autre. Contrairement aux chapitres

précédents, ces points vont être pondérés. Le poids d'un point sera une caractéristique de la machine correspondante. Cela pourra être sa mémoire, sa puissance de calcul ou bien la demande exprimée en nombre de requêtes émises par seconde.

A partir de ce nuage de points, nous pouvons définir le graphe de compatibilité, tel qu'il existe une arête entre 2 points si et seulement si leur distance est inférieure à un certain seuil.

Notre objectif sera alors de construire des cliques (ou à défaut des sous-graphes de très faible diamètre) et de poids supérieur (ou inférieur selon le contexte) à un certain seuil.

Ce travail a été réalisé dans le cadre de la thèse d'Hubert Larchevêque co-encadrée avec Olivier Beaumont et Philippe Duchon [BBDL11, BBL11b, Lar10a]

4.1 Introduction

Dans ce chapitre nous nous intéressons à deux problèmes d'agrégation de ressources : Bin Covering avec Contrainte de Distance (BCCD) et Bin Packing avec Contrainte de Distance (BPCD). Ces problèmes sont motivés par des applications de calcul volontaire et le placement de réplicats.

Le calcul volontaire (volunteer computing) a été popularisé par Seti@home : il s'agit d'utiliser les ressources de calcul inutilisées de centaines de milliers d'ordinateurs personnels dans le monde. La puissance agrégée de ces ressources peut atteindre plusieurs peta-Flops (comme c'est le cas pour le projet Folding@home), rivalisant ainsi avec les plus gros super calculateurs. Actuellement les seules applications capables de tourner sur ces plateformes sont des calculs constitués de petites tâches indépendantes. En vue d'étendre le potentiel applicatif de ces plateformes, nous nous sommes intéressés à des algorithmes permettant la fabrication de clusters de machines proches (en termes de latence), capables de traiter des tâches plus conséquentes (les données et les calculs étant distribués sur les membres du cluster). Les membres d'un même cluster étant amenés à échanger des messages, il est important que le diamètre des clusters reste petit ; le nombre de cluster quant à lui doit être le plus grand possible. Ce problème se ramène au problème du *bin covering avec contrainte de distance*.

La deuxième application visée est le placement de réplicats. Plusieurs variantes du problème de positionnement de serveurs ont été étudiés [PTW01, STA97, CW99, KPR00, KLL09, Chu06, KM07]. La plupart de ces problèmes considère un ensemble d'utilisateurs déployés dans un réseau (typiquement Internet) voulant accéder à un même service ou contenu. Le but est alors de déployer un ensemble (le plus petit possible) de serveurs à différents points du réseau capable de satisfaire la demande des utilisateurs. Le placement de ces serveurs est doublement contraint : d'une part la latence entre un serveur et les clients qui y sont associés doit rester faible. D'autre part, chaque client a une demande hétérogène (*i.e.* un nombre de requêtes par seconde propre à chaque client) et la somme des demandes des clients associées à un serveur doit être plus petite que la capacité du serveur. Ce problème se ramène au problème du *bin packing avec contrainte de distance*.

Pour ces deux problèmes nous avons suivi la même démarche : nous avons tout d'abord montré que les problèmes considérés étaient difficiles à approximer. Dans un deuxième temps nous avons proposé des algorithmes d'approximation avec augmentation de ressources. Pour que ces algorithmes puissent être utilisés dans les contextes considérés, ils doivent être combinés avec des outils de plongement (*i.e.* Vivaldi). Nous concluons ces recherches par des évaluations expérimentales des différentes combinaisons d'outils de plongement+algorithmes de covering/packing.

La suite de ce chapitre est organisée de la manière suivante. Dans la Section 4.2, nous introduisons les définitions des problèmes et nous donnons des résultats d'inapproximabilité. Dans la Section 4.3, nous présentons des algorithmes d'approximation pour BCCD dont certains sont dédiés à certaines métriques. Dans la Section 4.4, nous présentons un algorithme d'approximation pour BPCD. Enfin, dans la Section 4.5, nous présentons deux outils, Vivaldi et Séquoia pouvant être utilisés pour estimer les latences entre des machines reliées à Internet et nous présentons une évaluation expérimentale de l'utilisation combinée de nos algorithmes avec ces outils sur des données terrain.

4.2 Définitions et résultats d'inapproximation

4.2.1 Définitions et notations

Dans cette section nous présentons les définitions et notations nécessaires à nos 2 problèmes d'agrégation de ressources.

Soit S un ensemble et d une fonction de $d \times d \rightarrow \mathbb{R}$. On dit que (S, d) est un espace *semimétrique* si $\forall x, y \in S$ si : i) $d(x, y) \geq 0$ ii) $d(x, y) = 0$ si et seulement si $x = y$, iii) $d(x, y) = d(y, x)$. (S, d) est un espace ρ -*inframétrique* si (S, d) est un espace semi-métrique tel que $\forall x, y, z \in S$ $d(x, y) \leq \rho \max\{d(x, z), d(z, y)\}$.

Observons qu'un espace métrique est un espace semi-métrique qui vérifie l'inégalité triangulaire : $\forall x, y, z \in S$ $d(x, y) \leq d(x, z) + d(z, y)$. Observons également qu'un espace métrique est aussi un espace 2-inframétrique car $d(x, z) + d(z, y) \leq 2 \max\{d(x, z), d(z, y)\}$.

La *métrique Vivaldi* est définie de la manière suivante : chaque élément de l'espace possède un triplet de coordonnées dans $\mathbb{R}^2 \times \mathbb{R}^+$; la distance entre 2 éléments $a = (a_x, a_y, a_h)$ et $b = (b_x, b_y, b_h)$ est : $d(a, b) = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2} + a_h + b_h$. Une *métrique d'arbre* est une métrique telle que les éléments peuvent être plongés dans un arbre avec des arêtes pondérées tel que la distance entre 2 éléments est égale à la distance entre leurs images dans l'arbre.

Une instance \mathcal{I} de BPCD ou de BCCD peut être décrite à l'aide d'un 4-quadruplet $\mathcal{I} = (S, d, w, d_{\max})$, où S est un ensemble d'éléments $S = \{e_1, \dots, e_n\}$, (S, d) est un espace semi-métrique, w est une fonction de poids $w : S \rightarrow (0, 1)$ et d_{\max} une distance seuil. Nous ne considérons pas les éléments de poids supérieur à 1, car ces éléments peuvent être mis dans un sac de manière triviale pour les problèmes de bin covering et les problèmes de bin packing contenant de tels éléments n'admettent pas de solution.

Définition 4.1 (*BCCD : Bin covering avec contrainte de distance*). *Problème BCCD : étant donnée une instance*

$\mathcal{I} = (S, d, w, d_{\max})$, *trouver une collection de sous-ensembles deux à deux disjoints S_1, S_2, \dots, S_K de S de cardinalité maximale K telle que $\forall i \leq K$, $\sum_{e \in S_i} w(e) \geq 1$, $\forall (e_u, e_v) \in S_i$, $d(e_u, e_v) \leq d_{\max}$.*

On note par $\text{OPT}_{\text{BCCD}}(\mathcal{I})$ (ou simplement OPT_{BCCD}) la plus grande valeur de K pour une instance \mathcal{I} .

Pour l'instance de la Figure 4.1, $\text{OPT}_{\text{BCCD}} = 5$.

Définition 4.2 (*BPCD : Bin packing avec contrainte de distance*). *Problème BPCD : étant donnée une instance*

$\mathcal{I} = (S, d, w, d_{\max})$, *trouver une partition S_1, S_2, \dots, S_K de S de cardinalité minimale K telle que $\forall i \leq K$, $\sum_{e \in S_i} w(e) \leq 1$, $\forall (e_u, e_v) \in S_i$, $d(e_u, e_v) \leq d_{\max}$.*

On note par $\text{OPT}_{\text{BPCD}}(\mathcal{I})$ (ou simplement OPT_{BPCD}) la plus petite valeur de K pour une instance \mathcal{I} .

Pour l'instance de la Figure 4.1, $\text{OPT}_{\text{BPCD}} = 7$.

Il est possible de discrétiser notre espace semi-métrique et donc de se ramener à des graphes :

Définition 4.3 (*Graphe de Compatibilité*). *Le graphe de compatibilité $\text{Comp}(\mathcal{I}, d)$ associé à une instance \mathcal{I} est le graphe $G = (S; E)$ tel que $\forall u \in S, \forall v \in S$, $(u, v) \in E \Leftrightarrow d(u, v) \leq d$.*

Observons que $(e_i, e_j) \in E$ et $(e_j, e_k) \in E \Rightarrow d(e_i, e_k) \leq \rho d$ dans un espace ρ -inframétrique (si $\rho \leq 2$, l'espace est aussi un espace métrique). Notons que si S sont des points du plan Euclidien, $\text{Comp}(\mathcal{I}, 1)$ est le graphe de disques unitaires.

Notons que le BPCD est équivalent au problème *Bin Packing avec conflits* introduit dans [EL08] :

Définition 4.4 (*BPaC : Bin packing avec conflit*). *Problème BPaC : étant donnée un ensemble S , une fonction de poids w et un graphe (de conflit) $G = (S, E)$, trouver une partition S_1, S_2, \dots, S_K de S de cardinalité minimale K telle que $\forall i \leq K$, $\sum_{e \in S_i} w(e) \leq 1$, $\forall (e_u, e_v) \in S_i$, $(e_u, e_v) \notin E$.*

L'équivalence entre BPCD et BPaC se voit simplement en remarquant que le graphe de compatibilité de BPCD est le complémentaire de BPaC. Comme nous allons le voir par la suite le fait que BPCD utilise une fonction de distance va nous permettre de proposer des algorithmes d'approximation avec augmentation de ressources, chose qui n'a pas de sens pour les problèmes de BPaC.

Un algorithme \mathcal{A} est un (α, β) -algorithme d'approximation pour BCCD (resp. BPCD) s'il s'exécute en temps polynomial et construit des groupes de diamètre au plus βd_{\max} . De plus, le nombre de groupes qu'il construit est au moins (resp. au plus) αOPT^* , avec $\alpha \leq 1$, où OPT^* est le nombre de groupes d'une solution optimale contrainte à construire des groupes de diamètre d_{\max} .

4.2.2 Inapproximabilité avec faible augmentation de ressources

4.2.2.1 Inapproximabilité de BCCD

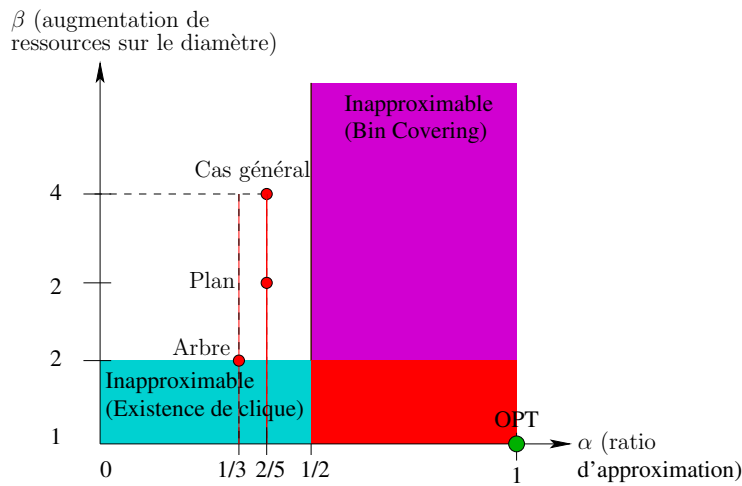


FIGURE 4.2 – Résultats d'approximabilités et d'inapproximabilités pour BCCD

Théorème 4.1. *Si $P \neq NP$, alors*

1. $\forall \varepsilon > 0$, et $\forall \alpha > 0$, il n'y a pas d' $(\alpha, (2 - \varepsilon))$ -algorithme d'approximation pour BCCD.
2. $\forall \varepsilon > 0$, et $\forall \beta > 1$, il n'y a pas d' $(1/2 - \varepsilon, \beta)$ -algorithme d'approximation pour BCCD.

Preuve 1) Pour le premier résultat d'inapproximabilité, nous allons considérer des instances où le poids des éléments est égal à $1/k$. Pour ces instances-là, décider si le problème du BCCD admet une solution avec au moins une bin revient à décider si le graphe de compatibilité de l'instance (qui rappelons-le, peut être n'importe quel graphe) admet une clique de taille k . Ce dernier problème étant NP-Complet, il n'existe pas d'algorithme d'approximation de BCCD si on ne relâche pas la contrainte sur les distances. De plus si on considère maintenant les instances telles que les distances entre chaque paire d'éléments est entière (c'est le cas par exemple si on considère la distance entre des sommets d'un graphe non-valué), le diamètre de chaque ensemble d'éléments est aussi entier. Donc toute bin B ayant un diamètre d'au plus $(2 - \varepsilon)$ a forcément un diamètre d'au plus 1. Donc pour ces instances l'utilisation d'une augmentation de ressources strictement inférieure à 2 ne modifie pas la complexité du problème.

2) Pour le deuxième résultat d'inapproximabilité, nous allons considérer des instances pour lesquelles la contrainte de distance peut-être ignorée ($\forall (e_1, e_2) \in S^2, d(e_1, e_2) < d_{\max}$). Pour ces instances tout (α, β) -algorithme d'approximation pour BCCD donne un algorithme d' α -approximation pour le problème du bin covering. Or le problème de bin covering ne peut être approximé (sauf si $P = NP$) à un facteur plus petit que $1/2$ (il suffit pour s'en convaincre de considérer les instances de bin covering se ramenant aux instances de 2 partitions : déterminer pour ces instances s'il est possible de construire

1 ou 2 bins est NP-Difficile). Ceci montre la deuxième partie du théorème. \square

4.2.2.2 Inapproximabilité de BPCD

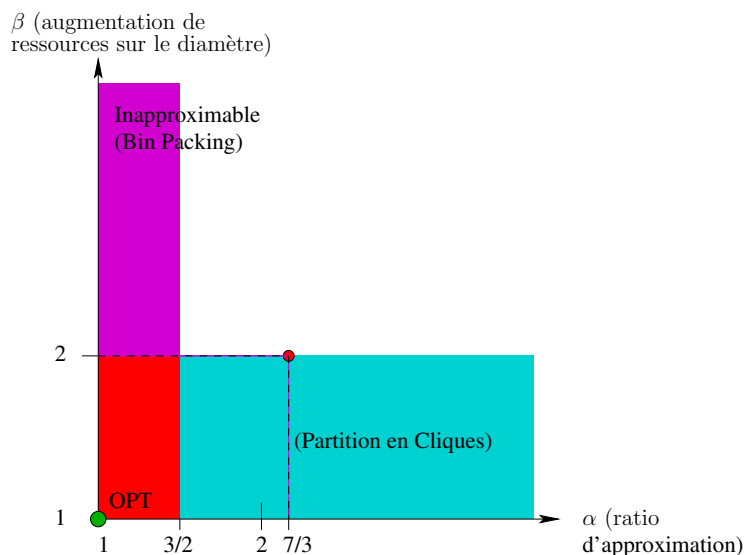


FIGURE 4.3 – Résultats d'approximabilités et d'inapproximabilités pour BPCD

Théorème 4.2. *Pour les instances où (S, d) est un espace métrique, si $P \neq NP$, alors*

1. $\forall \varepsilon > 0$, et $\forall \alpha > 1$, il n'y a pas d' $(\alpha, (2 - \varepsilon))$ -algorithme d'approximation pour BPCD.
2. $\forall \varepsilon > 0$, et $\forall \beta > 1$, il n'y a pas d' $(3/2 - \varepsilon, \beta)$ -algorithme d'approximation pour BPCD.

Preuve 1) Pour le premier résultat d'inapproximabilité, nous allons considérer des instances où le poids des éléments est négligeable ($\sum_{e \in S} w(e) < 1$). Pour ces instances-là chaque solution du problème est une partition en K cliques du graphe de compatibilité. De manière équivalente cela revient à K colorier le complémentaire du graphe de compatibilité. Or ce problème est difficile à approximer à un facteur $|V|^{1/7-\delta}$ pour tout $\delta > 0$ [BGS95]. Cela implique donc qu'il n'existe pas d'algorithme d'approximation (à un facteur d'approximation constant) de BPCD si on ne relâche pas la contrainte sur les distances. De plus si on considère maintenant les instances telles que les distances entre chaque paire d'éléments sont entières, le diamètre de chaque ensemble d'éléments est aussi entier. Donc toute bin B construit pour une telle instance ayant un diamètre d'au plus $(2 - \varepsilon)$ alors cette bin a un diamètre d'au plus 1. Donc pour ces instances l'utilisation d'une augmentation de ressources strictement inférieure à 2 n'aide pas pour le facteur d'approximation.

2) Pour le deuxième résultat d'inapproximabilité, nous allons considérer des instances pour lesquelles la contrainte de distance peut être ignorée ($\forall (e_1, e_2) \in S^2, d(e_1, e_2) < d_{\max}$). Pour ces instances tout (α, β) -algorithme d'approximation pour BPCD donne un algorithme d' α -approximation pour le problème du bin packing. Or le problème de bin packing ne peut être approximé (sauf si $P = NP$) à un facteur plus petit que $3/2$ (il suffit pour s'en convaincre de considérer les instances de bin packing se ramenant aux instances de 2 partitions : déterminer pour ces instances si 2 ou 3 bins sont nécessaires est NP-Difficile). Ceci montre la deuxième partie du théorème. \square

4.3 BCCD : Algorithmes d'approximations avec augmentation de ressources

4.3.1 Algorithme d'approximation pour des ρ -inframétriques

Nous présentons un $(\frac{2}{5}, \rho^2)$ -algorithme d'approximation pour toute instance plongée dans une ρ -inframétrique. Bien entendu, pour les instances plongées dans un espace métrique, cet algorithme est un $(\frac{2}{5}, 4)$ -algorithme d'approximation.

Cet algorithme utilise l'algorithme classique pour le bin covering, à savoir *Next-Fit-Decreasing*. Cet algorithme consiste à trier les éléments par ordre décroissant de poids. Puis les éléments sont ajoutés dans cet ordre dans la bin courante jusqu'à que le poids de la bin dépasse 1. A ce moment-là, la bin courante est fermée, et une nouvelle bin est ouverte. *Next-Fit-Decreasing* est un algorithme de 1/2-approximation pour le problème du bin covering et chaque bin de poids $1 + b$ ($b < 1$) contient seulement des éléments de poids strictement supérieur à b .

Algorithme 1 Algorithme en 2 phases gloutonnes pour BCCD

1: $U \leftarrow S$ // Éléments n'étant dans aucune bin.

2: $G = \text{Comp}(\mathcal{I}, d_{\max})$

Phase 1 :

1: **Pour** chaque élément e de S :

2: $C = U \cap \{e \cup N(e)\}$

3: Appliquer l'algorithme *Next-Fit-Decreasing* C

4: Supprimer les éléments placés dans des bins de U .

5: **Fin du Pour**

Phase 2 :

1: Appliquer la Phase 1 sur $G = \text{Comp}(\mathcal{I}, \rho d_{\max})$.

Théorème 4.3. *L'Algorithme 1 est un $(\frac{2}{5}, \rho^2)$ -algorithme d'approximation pour BCCD quand d est une ρ -inframétrique.*

Preuve Regardons dans un premier temps le diamètre des bins créées par l'Algorithme 1. Chaque élément d'une bin construite lors de la Phase 1 est à distance d'au plus d_{\max} d'un noeud central (le noeud e considéré dans la boucle), et donc le diamètre de cette bin est d'au plus ρd_{\max} . De manière analogue les éléments d'une bin construite en Phase 2 sont à distance ρd_{\max} d'un élément central. Donc le diamètre de ces bins est au plus $\rho^2 d_{\max}$.

Maintenant intéressons-nous au nombre de bins créées. Soit OPT une solution optimale et *économe*, c'est à dire qu'une bin de poids $1 + b$ ne peut contenir que des éléments de poids strictement supérieur à b . Soit $\mathcal{B}^{(k)}$, $k = 1, 2$ l'ensemble des bins construites à la Phase k . Nous appelons *zone étendue* l'ensemble des éléments $e \in S$ tels que $d(e; B) \leq d_{\max}$, où $d(e; B) = \min_{u \in B} d(e; u)$.

Tout d'abord remarquons que chaque bin B_{OPT} de OPT intersecte au moins une bin de $\mathcal{B}^{(1)}$. En effet, la Phase 1 se termine lorsqu'il n'est plus possible de créer de bin de rayon au plus d_{\max} . Or si une bin de OPT n'intersecte pas une bin de $\mathcal{B}^{(1)}$ elle pourrait être créée à la fin de la Phase 1. Notons au passage que chaque bin B_{OPT} est contenue dans la zone étendue d'une bin de $\mathcal{B}^{(1)}$.

Maintenant partitionnons les bins de $\mathcal{B}^{(1)}$ en 2 ensembles \mathcal{K}_1 et \mathcal{M}_1 , selon le nombre d'éléments des bins :

$$\left\{ \begin{array}{l} \mathcal{K}_1 = \{B_i^{(1)} \in \mathcal{B}^{(1)}, |B_i^{(1)}| = 2\}, \quad k_1 = |\mathcal{K}_1| \\ \text{et soit } \text{OPT}_K \text{ dénote le nombre de bins de } \text{OPT} \text{ qui} \\ \text{intersectent des bins de } \mathcal{K}_1, \\ \mathcal{M}_1 = \mathcal{B}^{(1)} \setminus \mathcal{K}_1, \quad m_1 = b_1 - k_1 \text{ et soit } \text{OPT}_M \text{ dénote} \\ \text{l'ensemble des bins de } \text{OPT} \setminus \text{OPT}_K \text{ qui intersecte les bins de } \mathcal{M}_1. \end{array} \right.$$

Remarque : il n'y a pas de bin contenant 1 seul élément car tous les éléments d'une instance ont un poids plus petit que 1.

Comme chaque bin de OPT intersecte un bin de Phase 1, on a $|\text{OPT}| \leq |\text{OPT}_K| + |\text{OPT}_M|$.

Comme chaque bin de \mathcal{K}_1 contient exactement 2 éléments, chaque bin de \mathcal{K}_1 intersecte au plus 2 bins de OPT_K . Donc $|\text{OPT}_K| \leq 2k_1$. Pour borner la cardinalité de OPT_M , nous introduisons w_{lost} , le poids des éléments non groupés par l'Algorithme 1 mais appartenant à des bins de OPT_M .

Lemme 4.1. $w_{\text{lost}} < m_1$

Preuve Dans la zone étendue de chaque bin de \mathcal{M}_1 , le poids des éléments non groupés par l'Algorithme 1 est strictement inférieur à 1 (sinon un bin de plus aurait été créé à la Phase 2. Donc $w_{\text{lost}} < m_1$. \square

De plus, $|\text{OPT}_M| \leq w(\text{OPT}_M) \leq w(\mathcal{M}) + w(\mathcal{B}^{(2)}) + w_{\text{lost}}$

Lemme 4.2. Si $B \in \mathcal{M}_1$, alors $w(B) < 1 + \frac{1}{|B|-1}$

Preuve Soit $w(B) = 1 + b$. Dans l'Algorithme 1, la stratégie Next-Fit-Decreasing est utilisée pour construire des bins, donc le poids de chaque élément est supérieur à b . Donc, $w(B) > b \times |B|$ et donc $1 + b > b \times |B|$. Cela implique que $b < \frac{1}{|B|-1}$. \square

Comme chaque bin B de \mathcal{M}_1 contient au moins 3 éléments, on a que $w(B) \leq \frac{3}{2}$. Par conséquent, $w(\mathcal{M}_1) < \frac{3}{2}m_1$. Puisque chaque bin de \mathcal{B}_2 a un poids d'au plus 2,

$$\begin{aligned} |\text{OPT}_m| &< \frac{3}{2}m_1 + 2b_2 + m_1 = \frac{5}{2}m_1 + 2b_2 \\ |\text{OPT}_{\text{BCCD}}| &< \frac{3}{2}m_1 + 2b_2 + 2k_1 < \frac{5}{2}(b_1 + b_2). \end{aligned}$$

Ceci conclut la preuve. \square

Comme le montre la Figure 4.4, la deuxième phase est nécessaire pour assurer un facteur d'approximation constant. En effet, il existe des instances pour lesquelles un algorithme glouton ne crée qu'une bin là où les solutions optimales en comptent un nombre arbitrairement grand.

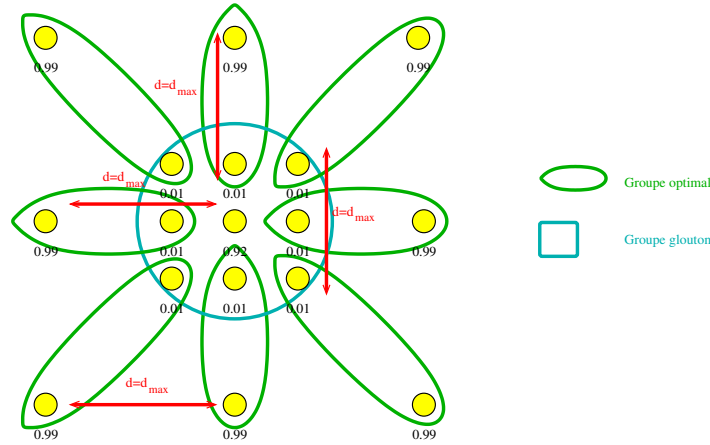


FIGURE 4.4 – Une fois le groupe central créé, on ne peut plus créer d'autres groupes.

4.3.2 $(\frac{2}{5}, 2 + \frac{3\sqrt{3}}{2})$ -algorithme d'approximation de BCCD pour un espace de Vivaldi

Nous présentons ici l'Algorithme 2, qui est un $(\frac{2}{5}, 2 + \frac{3\sqrt{3}}{2})$ -algorithme d'approximation lorsque les éléments sont plongés dans un espace de Vivaldi. Cet algorithme construit des bins de diamètre au plus

$\frac{\sqrt{3}+1}{2}d_{\max}$ lors de la Phase 1 puis de taille $\frac{\sqrt{3}+1}{2}(2 + \frac{\sqrt{3}+1}{2})d_{\max} = 2 + \frac{3\sqrt{3}}{2}d_{\max}$ lors de la Phase 2 (au lieu de ρd_{\max} et $\rho^2 d_{\max}$ pour le cas général).

Pour pouvoir décrire cet algorithme, nous avons besoin d'introduire la définition suivante :

Définition 4.5 (*Lentille de e_i et e_j*). Soit e_i et e_j 2 points. La lentille (symétrique) de e_i et e_j , notée $Le(e_i, e_j)$ est l'ensemble des points e_k tels que $d(e_i, e_k) \leq d(e_i, e_j)$ et $d(e_j, e_k) \leq d(e_i, e_j)$. On appelle $d(e_i, e_j)$ le diamètre de la lentille.

L'Algorithme 2 est paramétré par β' , un paramètre qui dépend de la métrique considérée. Pour l'espace de Vivaldi, $\beta' = \frac{\sqrt{3}+1}{2}$.

Algorithme 2 ($(\frac{2}{5}, \beta'(2 + \beta'))$)-algorithme d'approximation de BCCD.

- 1: $U \leftarrow S$
- 2: $G = \text{Comp}(\mathcal{I}, \beta' d_{\max})$.
- 3: $d \leftarrow d_{\max}$

Phase 1 :

- 1: **Tant que** qu'il existe une lentille de diamètre au plus d contenant une clique de G de poids d'au moins 1 :
- 2: Appliquer Next-Fit-Decreasing sur cette clique.
- 3: Supprimer les éléments placés dans des bins de U .
- 4: **Fin du Tant que**

Phase 2 :

- 1: $G = \text{Comp}(\mathcal{I}, \beta'(\beta' + 2)d_{\max})$.
 - 2: $d \leftarrow (\beta' + 2)d_{\max}$
 - 3: Appliquer la Phase 1.
-

Comme chaque clique de diamètre d_{\max} est incluse dans une lentille de diamètre au plus d_{\max} , construire les cliques de diamètre d_{\max} et de poids supérieur à 1 se fait simplement en examinant toutes les lentilles définies à partir d'éléments de S (soit $O(n^2)$ lentilles), et identifier dans chaque lentille les bins pouvant être construites.

Toute la difficulté maintenant est de déterminer si une lentille contient une clique de poids ≥ 1 . Déterminer si un graphe contient une clique de poids ≥ 1 est a priori NP-Complet, mais il devient polynomial lorsque le graphe peut être décomposé en 2 cliques. Comme le montre le lemme suivant, c'est bien le cas pour des valeurs de β' bien choisies :

Lemme 4.3. *Étant donnés 2 éléments $e_i, e_j \in S^2$, dans une lentille $Le(e_i, e_j)$ de diamètre au plus d_{\max} , on définit le graphe $G_{e_i, e_j} = \text{Comp}(\mathcal{I}, \beta' d_{\max}) \cap Le(e_i, e_j)$. Les sommets du graphe G_{e_i, e_j} peuvent être partitionnés en 2 cliques pour $\beta' = \frac{\sqrt{3}+1}{2}$ pour l'espace Vivaldi, et $\beta' = 1$ pour le plan Euclidien.*

L'analyse des solutions calculées par l'Algorithme 2 est exactement la même que celle de l'Algorithme 1.

Théorème 4.4. *L'Algorithme 2 est un $(\frac{2}{5}, 2 + \frac{3\sqrt{3}}{2})$ -algorithme d'approximation de BCCD pour l'espace Vivaldi et un $(\frac{2}{5}, 3)$ -algorithme d'approximation de BCCD pour le plan Euclidien.*

Remarque : bien que l'algorithme dédié à la métrique de Vivaldi soit moins performant dans le pire des cas que l'algorithme général : le diamètre des bins créées par l'algorithme dédié peut être plus grand que celui généré par l'algorithme général, celui-ci s'avère en pratique aussi efficace : le diamètre moyen des bins créées est sensiblement égal à celui du diamètre moyen des bins créées par l'algorithme générique comme nous le verrons dans la Section 4.5.

4.3.3 Métrique d'arbre

Nous présentons ici l'Algorithme 3, qui est un algorithme d'approximation lorsque la métrique considérée est une métrique d'arbre. Par la suite nous supposons que les éléments de l'instance sont plongés sur un arbre T définissant la métrique. De plus nous enracinons (de manière arbitraire) cet arbre.

Contrairement aux algorithmes que nous venons de présenter, l'Algorithme 3 n'utilise pas l'algorithme Next-Fit-Decreasing, mais une variante de cet algorithme que nous appelons *Next-Fit-Distance-Decreasing* : étant donné un noeud de référence, l'algorithme commence par trier les éléments par distance décroissante à ce noeud. Les éléments sont ensuite ajoutés dans la bin ouverte dans cet ordre jusqu'à ce que le poids de cette bin dépasse 1. A ce moment-là, la bin est fermée et une nouvelle bin est ouverte.

Étant donné un noeud de l'arbre T , nous appelons $parent(x)$ le parent de x dans T et $children(x)$ l'ensemble des enfants de x . La racine de T est noté r .

Avant de décrire l'algorithme, nous devons introduire la définition suivante :

Définition 4.6. *Étant donné un arbre T et un noeud $e \in T$, $SUBTREE_{d_{\max}}(e)$ est le sous-arbre de T enraciné en e tel que $\forall e' \in SUBTREE_{d_{\max}}(e), d(e, e') \leq d_{\max}$.*

Algorithme 3 $(\frac{1}{3}, 2)$ -approximation algorithme pour BCCD dans une métrique d'arbre

1: $U \leftarrow S$

2: BUILDINSUBTREE(r)

Procédure BUILDINSUBTREE(e) :

1: **Pour** chaque $e' \in children(e)$:

2: BUILDINSUBTREE(e')

3: **Fin du Pour**

4: Appliquer Next-Fit-Distance-Decreasing sur $U \cap SUBTREE_{d_{\max}}(e)$

5: Supprimer les éléments placés dans des bins de U .

Théorème 4.5. *L'Algorithme 3 est un $(\frac{1}{3}, 2)$ -algorithme d'approximation pour BCCD où d est une métrique d'arbre.*

4.4 Algorithmes d'approximations pour BPCD

Nous présentons ici un $(\frac{7}{3}, \rho)$ -algorithme d'approximation de BPCD lorsque d est une ρ -inframétrique. Cet algorithme est une adaptation de l'algorithme proposé par Epstein et Levin [EL08]. L'algorithme proposé dans [EL08] est un algorithme d'approximation pour le problème du bin packing avec conflit qui fonctionne uniquement pour des graphes de conflits pour lesquels le problème d'extension de précoloration peut être résolu optimalement en temps polynomial.

L'algorithme que nous présentons utilise l'algorithme d'approximation classique pour le bin packing, à savoir *First-Fit-Decreasing*. Cet algorithme consiste à trier les éléments par ordre décroissant de poids. Puis les éléments sont ajoutés dans cet ordre dans la première bin ayant assez de place pour le recevoir. Si aucune bin déjà ouverte n'est capable de le recevoir alors une nouvelle bin est ouverte. Cet algorithme a un facteur d'approximation de $11/9$ [UI71].

Avant de présenter notre adaptation, rappelons tout d'abord quelques définitions issues de [EL08]. Le *poids étendu* d'un élément x , noté $e(x)$ est défini de la manière suivante : si $w(x) > \frac{1}{2}$, alors $e(x) = 1$; si $w(x) \in \mathcal{I}_j = (\frac{1}{j+1}, \frac{1}{j}]$ pour un entier $j > 1$, alors $e(x) = w(x) + \frac{1}{j(j+1)}$.

Théorème 4.6 (Reformulation du Théorème 12 de [EL08]). *Soit m_1 le nombre de bins faites par l'Algorithme 4 lors de la boucle tant que. Si $m_1 + |M| \leq OPT_{BPCD}$ et si chaque ensemble de M a un diamètre d'au plus ρd_{\max} , alors l'Algorithme 4 est un $(\frac{7}{3}, \rho)$ -algorithme d'approximation de BPCD.*

Algorithme 4 $(\frac{7}{3}, \rho)$ -algorithme d'approximation pour BPCD

- 1: $U \leftarrow S$
 - 2: $G = \text{Comp}(\mathcal{I}, d_{\max})$
 - 3: **Tant que** il y a dans G un triangle $\{a, b, c\}$ tel que $w(a) + w(b) + w(c) \leq 1$, $e(a) + e(b) + e(c) > 1$ et $w(c) \leq w(b) \leq w(a) \leq \frac{1}{2}$ ou bien une arête $\{a, b\}$ telle que $w(a) + w(b) \leq 1$ et $e(a) + e(b) > 1$:
 - 4: Parmi ces ensemble, choisir celui de poids étendu maximal et faire une bin avec cet ensemble.
 - 5: Supprimer ces éléments de U .
 - 6: **Fin du Tant que**
 - 7: Construire une partition M de U
 - 8: Appliquer First-Fit-Decreasing dans chaque ensemble de M .
-

Toute la difficulté maintenant réside dans le calcul d'une partition $|M|$ qui satisfasse les conditions du Théorème 4.6.

Ceci peut être réalisé grâce à l'Algorithme 5 :

Algorithme 5 Construction d'une partition M de U en ensembles de diamètre au plus ρd_{\max}

- 1: Construire $\text{Comp}'(\mathcal{I}, d_{\max})$, le sous-graphe de $\text{Comp}(\mathcal{I}, d_{\max})$ dans lequel on a supprimé toutes les arêtes joignant 2 sommet de poids $\geq \frac{1}{2}$
 - 2: Construire un ensemble maximal indépendant M de $\text{Comp}'(\mathcal{I}, d_{\max})$ tel que tous les éléments de poids $\geq 1/2$ soient dans cet ensemble.
 - 3: Associer arbitrairement les éléments de $U \setminus M$ à un de leurs voisins dans M . Retourner la partition ainsi engendrée.
-

Lemme 4.4. La partition M de U calculée par l'Algorithme 5 satisfait $|M| \leq \text{OPT}_{\text{BPCD}} - m_1$.

Théorème 4.7. L'Algorithme 4 combiné avec l'Algorithme 5 est un $(\frac{7}{3}, \rho)$ -algorithme d'approximation pour BPCD quand d est une ρ -inframétrie.

Ce théorème est une conséquence directe du Théorème 4.6 et Lemme 4.4. L'Algorithme 4 dans le cas où d est une métrique est optimal par rapport à l'augmentation de ressources, dans le sens où il n'existe pas d'algorithme d'approximation utilisant une augmentation de ressources plus faible que 2.

4.5 Vers une utilisation dans les plateformes à grande échelle

Dans cette section nous étudions l'utilisation de deux outils de plongement d'Internet pour l'agrégation de ressources dans des plateformes de grande échelle. Les outils de plongement d'Internet sont utilisés pour placer dans un espace simple (généralement un espace métrique) des ressources distribuées connectées via Internet. Vivaldi [DCKM04a] et Sequoia [RMK⁺09a] sont de tels outils de plongement parmi les plus rencontrés dans la littérature. Ces outils assignent à chaque ressource (noeud) une position dans un espace simple, de telle sorte que la distance entre n'importe quelle paire de noeuds dans cet espace soit une bonne approximation de la distance réelle les séparant (leur latence) dans Internet. Une alternative évidente pourrait être de mesurer toutes les latences et de travailler sur la matrice complète de latences. Dans le contexte de plateformes à grande échelle qui nous intéresse, cette approche n'est pas réaliste, en raison du coût nécessaire pour effectuer ces mesures, et de la dynamique intrinsèque à ce type de plateformes.

L'utilisation d'outils de plongement provoque une (légère) distorsion des latences, mais en contrepartie permet de travailler dans des espaces plus simples, et ainsi d'élaborer des algorithmes d'approximation efficaces. Dans le cadre d'une application spécifique, comme l'agrégation de ressources, seule la performance du couple (plongement, algorithme d'approximation) est plus important que la qualité de chaque algorithme pris séparément.

4.5.1 Latence et Outils de plongements

Dans cette section, nous présentons certaines caractéristiques de l'espace des latences d'Internet et deux des outils de plongement les plus répandus dans la littérature : Vivaldi et Sequoia.

L'espace des latences d'Internet n'est pas un espace métrique, mais seulement un espace semi-métrique, car il contient un grand nombre de violations de l'inégalité triangulaire [LBSB09b]. Ces violations d'inégalités triangulaires viennent par exemple de la charge du réseau ou de la configuration des routeurs qui font que le plus court chemin n'est pas forcément celui emprunté par les paquets. Cependant, comme le montre ces travaux [LFV08], l'espace des latences d'Internet semble être raisonnablement proche d'un espace ρ -inframétrique. Plus précisément les résultats [LFV08] montrent que 99.5% des triplets vérifient cette inégalité pour $\rho = 2$.

L'espace des latences d'Internet n'étant pas métrique, il ne peut être plongé dans un espace métrique sans subir de perte d'informations. Sequoia et Vivaldi permettent cependant une bonne approximation des latences entre chaque paire de noeuds [DCKM04a].

Vivaldi, dans sa version standard, associe à chaque noeud d'un réseau un point dans l'espace de métrique Vivaldi (voir la définition dans la Section 4.2).

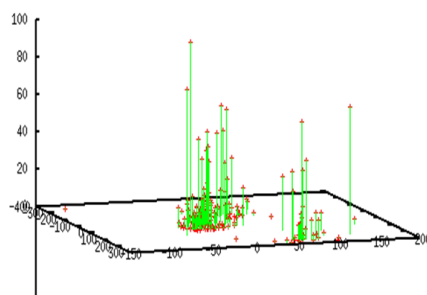


FIGURE 4.5 – Plongement dans l'espace Vivaldi de 226 noeuds de Planetlab

Sequoia plonge les noeuds d'Internet dans un ou plusieurs arbres pondérés dans lesquels chaque noeud est soit une feuille soit la racine, et dans lesquels les noeuds internes sont des noeuds virtuels. La distance entre deux points dans le réseau d'origine est bien approchée par la distance dans l'arbre de plongement (ou par la médiane des distances dans le cas de plusieurs arbres). Comme présenté dans [RMK⁺09a], la précision des prédictions obtenues en utilisant Sequoia est meilleure dès que quelques arbres sont utilisés.

4.5.2 Évaluation expérimentale

Nous présentons ici des couples algorithme d'approximation pour BCCD/outil de plongement. Nous avons également effectué une évaluation similaire des couples algorithmes d'approximation pour BPCD/outil de plongements [BBL11b]. Les méthodes et conclusions étant similaires, nous ne présentons ici que la partie BCCD.

Nous utilisons un jeu de données réel pour estimer les latences, issu du projet Meridian¹, contenant toutes les mesures de latence relatives à un ensemble de 2500 serveurs DNS mesurés grâce à la méthode King [GSG02]. Pour estimer les capacités hétérogènes des ressources, nous nous appuyons sur un jeu de données du projet XtremLab². XtremLab consiste en une liste des caractéristiques des noeuds utilisés par des applications BOINC. Nous en extrayons les données correspondant aux capacités de calcul mises

1. <http://www.cs.cornell.edu/People/egs/meridian/data.php>

2. <http://xw01.lri.fr:4320/>

à disposition par chaque noeud sur la plateforme (valeur exprimée en Flops), obtenant une puissance totale de 3632 GFlops.

Partant de ces données nous effectuons le plongement de la matrice de latences en utilisant soit Vivaldi soit Sequoia (avec seulement un arbre pour Sequoia). Nous appliquons à chacun de ces plongements l'algorithme correspondant pour BCCD, le tout sur le même ensemble d'éléments pondérés, et pour différentes valeurs de d_{\max} , allant de la latence minimale entre deux points, préalablement identifiée, à une valeur de $200ms$ (au-delà de cette valeur, aucune évolution n'est observée).

Nous avons effectué des simulations pour différentes valeurs de W , et avons choisi $W \simeq 11GFlops$ comme étant la valeur la plus représentative (chaque groupe doit donc agréger environ 0.3% du poids total). Pour de plus petites valeurs de W , les solutions tendent à placer de moins en moins d'éléments dans chaque groupe, puisque les poids des éléments sont plus proches de la valeur seuil. Le nombre de groupes construits tend alors vers une valeur stable pour une plus petite valeur de d_{\max} . En utilisant de plus petites valeurs pour W , la contrainte de poids devient la plus importante et BCCD se rapproche d'un problème de *bin covering* classique, puisque la contrainte de distance est de moins en moins sensible : même quand d_{\max} augmente, la construction de groupes peut toujours s'avérer difficile en raison du manque de poids. En utilisant cette valeur de W , un groupe contient en moyenne environ 7 éléments.

La Figure 4.6 présente à gauche le nombre de groupes valides, et à droite le diamètre moyen des groupes construits pour différentes valeurs de d_{\max} , en utilisant chacune des combinaisons (plongement, algorithme dédié) que nous avons sélectionnées. Dans tous les cas, l'Algorithme 1 appliqué à la matrice

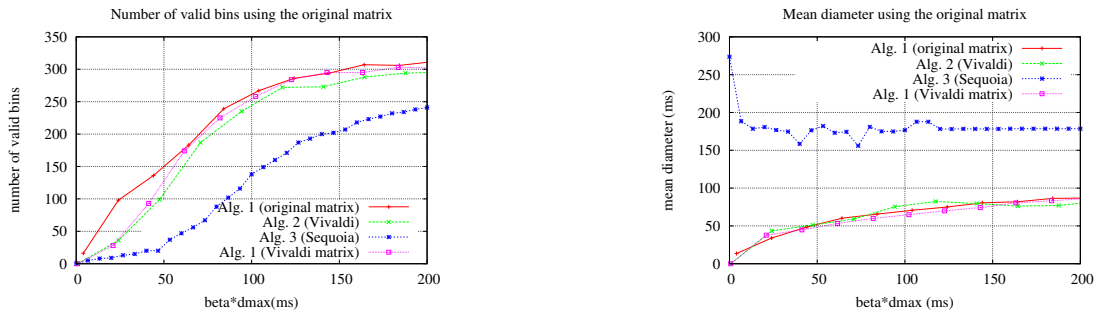


FIGURE 4.6 – Nombre de groupes valides (gauche) et diamètre moyen des groupes construits (droite) pour BCCD, pour différentes valeurs de d_{\max} , en utilisant différents outils de plongement.

des latences offre de meilleures performances à la fois que l'Algorithme 2 et que l'Algorithme 1 utilisant la matrice de latences obtenue à partir du plongement dans un espace Vivaldi. Tous ces algorithmes ont de meilleures performances que l'Algorithme 3.

L'Algorithme 2 et l'Algorithme 1 utilisant la matrice de latences obtenue à partir du plongement dans Vivaldi ont des performances très comparables. Il est intéressant de constater que, bien que l'Algorithme 2 requière une augmentation de ressources de $\beta \simeq 4.6$, le diamètre moyen des groupes qu'il construit tend à être équivalent au diamètre moyen des groupes construits par l'Algorithme 1, qui lui requiert seulement une augmentation de ressources de $\beta = 4$.

L'utilisation de Vivaldi offre des performances quasi équivalentes à l'algorithme dédié au cas général, malgré la perte de précision dans la prédiction des distances induite par le plongement. Ainsi cette perte de précision due au plongement semble bien compensée par l'élaboration d'un algorithme efficace dédié à un espace métrique plus simple. Par ailleurs, cela montre qu'utiliser un outil de plongement conjointement à un algorithme dédié à celui-ci permet d'obtenir des résultats aussi bons qu'en utilisant une matrice de latences complète, alors qu'obtenir une telle matrice de latences est, en pratique, impossible.

4.6 Conclusion et perspectives

Nous avons présenté ici quelques résultats d'inapproximabilité ainsi que quelques algorithmes d'approximation avec augmentation de ressources pour les problèmes de BCCD et BPCD. Nous avons ensuite essayé de mesurer l'efficacité effective de ces algorithmes d'agrégation lorsqu'ils étaient combinés avec des outils de plongement d'Internet : Vivaldi et Séquoia. Ces derniers résultats ont montré que nos algorithmes se mariaient relativement bien avec Vivaldi et que Sequoia semblait moins robuste (même avec l'utilisation de plusieurs arbres au lieu d'un seul comme présenté dans la section précédente).

Les algorithmes d'approximations que nous avons présentés sont centralisés et ont généralement une complexité surlinéaire, ce qui est rédhibitoire pour les applications visées. Notons que pour le problème de BCCD, nous avons proposé [BBDL08c, BBED⁺08b] un $(\frac{1}{3}, 4)$ -algorithme d'approximation distribué fonctionnant en temps $O(\log n)$ dans le cas où l'espace considéré est le plan muni de la métrique L_∞ .

Pour ce qui est de l'inapproximabilité des problèmes de BCCD et BPCD nous avons obtenu des résultats sur des instances où l'on pouvait soit négliger la contrainte de distance soit négliger la contrainte de poids. Il serait intéressant d'avoir (ou non) des résultats d'inapproximabilité mettant en évidence que la combinaison des 2 contraintes est plus difficile que la juxtaposition de ces contraintes. Notons que c'est déjà le cas pour le facteur d'approximation asymptotique. Pour le problème du bin packing il n'existe pas d'algorithme d'approximation à un facteur meilleur que $3/2$ (sauf si $P = NP$), mais il existe un schéma asymptotique d'approximation en temps polynomial [dVVL81] (APTAS) : c'est à dire que pour tout ε il existe un algorithme polynomial A produisant des solutions de taille au plus $(1 + \varepsilon) \cdot OPT + O(1)$ où OPT est le nombre optimal de bin. Dans le cas du BPCD il n'est pas possible d'avoir un APTAS (sauf si $P = NP$). En effet, en utilisant la contrainte de distance on peut construire des instances de BPCD se ramenant à $n/2$ problèmes de bin packing indépendants, pour lesquels il est NP-difficile de déterminer si 2 ou 3 bins sont nécessaires.

Dans la thèse d'Hubert Larchevêque, nous avons suivi la démarche suivante : nous avons identifié 2 problèmes pratiques : agrégation de ressources pour du calcul volontaire ou pour du placement de réplicats. Nous avons ensuite modélisé ces problèmes de manière formelle (BCCD et BPCD). Nous avons alors montré que ces problèmes étaient difficiles à résoudre et même à approximer et nous avons proposé des algorithmes avec des garanties de performances. Puis nous avons considéré ces problèmes théoriques dans un contexte pratique, c'est à dire en utilisant des outils de plongement.

Dans le cadre de la thèse de Przemysław Uznański que nous co-encadrons avec Olivier Beaumont et Lionel Eyraud-Dubois, nous avons repris cette démarche pour aborder un autre problème : la diffusion (le broadcast) sur internet. Pour ce type d'application où de gros volumes de données sont échangés, la latence peut-être ignorée et la bande-passante disponible est le paramètre déterminant.

Pour ce problème nous avons considéré un modèle de communication à la fois réaliste et instanciable : le modèle *last-mile* en présence de noeuds gardés. Dans ce modèle, chaque noeud est caractérisé par une bande-passante entrante et une bande passante sortante. Un noeud est dit *gardé* s'il se trouve derrière un pare-feu ou un routeur NAT. Deux noeuds gardés ne peuvent communiquer entre eux car aucun des deux ne peut initier la communication. En plus d'être réaliste, ce modèle est instanciable. En effet, il existe des outils permettant (à l'instar de Vivaldi pour la latence) de déterminer les paramètres du modèle pour une instance de réseau [BEDW11]. Nous avons montré [BBEDU11] qu'un schéma de diffusion atteignant un débit optimal pouvait nécessiter un degré arbitrairement grand (i.e. certains noeuds devant communiquer avec presque tous les autres noeuds). De plus atteindre le débit optimal avec des contraintes de degré strictes rendait le problème NP-Difficile. Parallèlement à cela nous avons montré qu'un schéma atteignant un débit d'au moins $5/7$ fois le débit optimal pouvait être construit tout en gardant le degré maximum bas.

Les outils de prédiction de bandes passantes et de latences ne sont pas pleinement satisfaisants et il reste de la place pour des outils offrant des prédictions plus précises. Après avoir étudié les 2 types d'outils nous pensons qu'un outil disposant de mesures de bandes passantes et de latences pourrait être plus efficace pour estimer les latences et les bandes passantes que n'importe quel outil traitant ces informations séparément.

Bibliographie

- [ABP06] E. Ackerman, G. Barequet, and R. Y. Pinter. A bijection between permutations and floorplans, and its applications. *Discrete Applied Math.*, 154(12) :1674–1685, 2006.
- [ADDJ90] Ingo Althöfer, Gautam Das, David P. Dobkin, and Deborah Joseph. Generating sparse spanners for weighted graphs. In *SWAT*, pages 26–37, 1990.
- [AFL09] Luca Aleardi, Éric Fusy, and Thomas Lewiner. Schnyder woods for higher genus triangulated surfaces, with applications to encoding. *Discrete and Computational Geometry*, 42 :489–516, 2009. 10.1007/s00454-009-9169-z.
- [B97] Miklós Bóna. Exact enumeration of 1342-avoiding permutations : A close link with labeled trees and planar maps. *Journal of Combinatorial Theory, Series A*, 80(2) :257 – 272, 1997.
- [Bax64] G. Baxter. On fixed points of the composite of commuting functions. *Proceedings of the American Mathematical Society*, 15 :851–855, 1964.
- [Bax01] R. J. Baxter. Dichromatic polynomials and Potts models summed over rooted maps. *Ann. Comb.*, 5(1) :17–36, 2001.
- [BB09] O. Bernardi and N. Bonichon. Catalan’s intervals and realizers of triangulations. *JCT-A*, 116(1) :55–75, 2009.
- [BBDL08c] O. Beaumont, N. Bonichon, Ph. Duchon, and H. Larchevêque. Distributed approximation algorithm for resource clustering. In Alexander A. Shvartsman and Pascal Felber, editors, *SIROCCO 2008*, volume 5058 of *Lecture Notes in Computer Science*, pages 61–73. Springer, 2008.
- [BBDL11] Olivier Beaumont, Nicolas Bonichon, Philippe Duchon, and Hubert Larchevêque. Use of Internet Embedding Tools for Heterogeneous Resources Aggregation. In *HCW 2011*, pages 114–124, 2011.
- [BBED⁺08b] O. Beaumont, N. Bonichon, L. Eyraud-Dubois, Ph. Duchon, and H. Larchevêque. A distributed algorithm for resource clustering in large scale platforms. In *OPODIS 2008*, volume 5401 of *Lecture Notes in Computer Science*, pages 564–567. Springer, 2008.
- [BBEDU11] Olivier Beaumont, Nicolas Bonichon, Lionel Eyraud-Dubois, and P. Uznanski. Broadcasting on Large Scale Heterogeneous Platforms with connectivity artifacts under the Bounded Multi-Port Model. In *ICPADS 2011 – 17th International Conference on Parallel and Distributed Systems*, volume Proceeding of ICPADS 2011, Taïnan, Taïwan, Province De Chine, 2011.
- [BBL11b] Olivier Beaumont, Nicolas Bonichon, and Hubert Larchevêque. Modeling and Practical Evaluation of a Service Location Problem in Large Scale Networks. In *ICPP 2011*, page 482–491, 2011.
- [BBMF10] Nicolas Bonichon, Mireille Bousquet-Mélou, and Éric Fusy. Baxter permutations and plane bipolar orientations. *Séminaire Lotharingien de Combinatoire*, 61A, 2010.
- [BCCS08] Prosenjit Bose, Paz Carmi, Sébastien Collette, and Michiel Smid. On the stretch factor of convex delaunay graphs. In *19th Annual International Symposium on Algorithms and Computation (ISAAC)*, volume 5369 of *Lecture Notes in Computer Science*, pages 656–667. Springer, December 2008.

- [BCCY12] Prosenjit Bose, Paz Carmi, and Lilach Chaitman-Yerushalmi. On bounded degree plane strong geometric spanners. *Journal of Discrete Algorithms*, 15(0) :16 – 31, 2012.
- [BDL⁺11] Prosenjit Bose, Luc Devroye, Maarten Löffler, Jack Snoeyink, and Vishal Verma. Almost all Delaunay triangulations have stretch factor greater than $\pi/2$. *Computational Geometry*, 44(2) :121–127, 2011.
- [BEDW11] O. Beaumont, L. Eyraud-Dubois, and Y. Won. Using the last-mile model as a distributed scheme for available bandwidth prediction. In *Proceedings of the EuroPar 2011 conference*, 2011.
- [Ber07] O. Bernardi. Bijective counting of tree-rooted maps and shuffles of parentheses systems. *Electronic J. Combinatorics*, 14(1), 2007. Paper R9.
- [BFM07] N. Bonichon, S. Felsner, and M. Mosbah. Convex drawings of 3-connected planar graphs. *Algorithmica*, 47(4) :399–420, 2007.
- [BFvRV12] Prosenjit Bose, Rolf Fagerberg, André van Renssen, and Sander Verdonschot. Competitive routing in the half-theta₆-graph. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 1319–1328. SIAM, 2012.
- [BGH05] N. Bonichon, C. Gavoille, and N. Hanusse. Canonical decomposition of outerplanar maps and application to enumeration, coding and generation. *Journal of Graph Algorithms and Applications*, 9(2) :185 – 204, 2005.
- [BGH⁺06] N. Bonichon, C. Gavoille, N. Hanusse, D. Poulalhon, and G. Schaeffer. Planar graphs, via well-orderly maps and trees. *Graphs and Combinatorics*, 22 :1–18, 2006.
- [BGHI10a] Nicolas Bonichon, Cyril Gavoille, Nicolas Hanusse, and David Ilcinkas. Connections between theta-graphs, delaunay triangulations, and orthogonal surfaces. In *WG'2010*, volume 6410 of *Lecture Notes in Computer Science*, pages 266–278, 2010.
- [BGHP10] Nicolas Bonichon, Cyril Gavoille, Nicolas Hanusse, and Ljubomir Perković. Plane spanners of maximum degree six. In *ICALP'10*, volume 6198 of *Lecture Notes in Computer Science*, pages 19–30, 2010.
- [BGHP12] Nicolas Bonichon, Cyril Gavoille, Nicolas Hanusse, and Ljubomir Perković. The stretch factor of L_1 - and L_∞ -Delaunay triangulations. In L. Epstein and P. Ferragina, editors, *20th Annual European Symposium on Algorithms (ESA)*, volume 7501 of *Lecture Notes in Computer Science*, pages 205–216. Springer-Verlag, September 2012.
- [BGKN07] M. Bodirsky, O. Giménez, M. Kang, and M. Noy. Enumeration and limit laws of series-parallel graphs. *Europ. J. Combinatorics*, 28(8) :2091–2105, 2007.
- [BGS95] M. Bellare, O. Goldreich, and M. Sudan. Free bits, pcps and non-approximability-towards tight results. *Foundations of Computer Science, Annual IEEE Symposium on*, 0 :422, 1995.
- [BGS05] Prosenjit Bose, Joachim Gudmundsson, and Michiel Smid. Constructing plane spanners of bounded degree and low weight. *Algorithmica*, 42(3-4) :249–264, 2005.
- [BLM02] N. Bonichon, B. Le Saëc, and M. Mosbah. Wagner's theorem on realizers. In *Proceedings of International Colloquium on Automata, Languages and Programming 2002 (ICALP'02)*, volume 2380 of *Lecture Notes in Computer Science*, pages 1043–1053. Springer-Verlag, 2002.
- [BM03] M. Bousquet-Mélou. Four classes of pattern-avoiding permutations under one roof : generating trees with two labels. *Electronic J. Combinatorics*, 9(2) :Research Paper 19, 2003.
- [BMSU01] Prosenjit Bose, Pat Morin, Ivan Stojmenović, and Jorge Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Wireless Networks*, 7 :609–616, 2001. 10.1023/A :1012319418150.
- [Bon05] N. Bonichon. A bijection between realizers of maximal plane graphs and pairs of non-crossing dyck paths. *Discrete Mathematics*, 298 :104–114, 2005. FPSAC'02 Special Issue.

- [BS] P. Bose and M. Smid. On plane geometric spanners : A survey and open problems. submitted.
- [BSX09] P. Bose, M. Smid, and D. Xu. Delaunay and diamond triangulations contain spanners of bounded degree. *International Journal of Computational Geometry and Applications*, 19(2), 2009.
- [BTV99] G. Di Battista, R. Tamassia, and L. Vismara. Output-sensitive reporting of disjoint paths. *Algorithmica*, 23(4) :302–340, 1999.
- [CAFL08] Luca Castelli Aleardi, Éric Fusy, and Thomas Lewiner. Schnyder woods for higher genus triangulated surfaces. In *24st Annual ACM Symposium on Computational Geometry (SoCG)*, pages 311–319, June 2008.
- [CD85] P. Chew and R. L. Dyrsdale. Voronoi diagrams based on convex distance functions. In *SCG '85 : Proceedings of the first annual symposium on Computational geometry*, pages 235–244. ACM, 1985.
- [CDV86] R. Cori, S. Dulucq, and G. Viennot. Shuffle of parenthesis systems and Baxter permutations. *J. Combin. Theory Ser. A*, 43(1) :1–22, 1986.
- [CGHK78] F. R. K. Chung, R. L. Graham, V. E. Hoggatt, Jr., and M. Kleiman. The number of Baxter permutations. *J. Combin. Theory Ser. A*, 24(3) :382–394, 1978.
- [Cha06] Frédéric Chapoton. Sur le nombre d’intervalles dans les treillis de tamari. *Séminaire Lotharingien de Combinatoire*, 55f, 2006.
- [Cha11] Guillaume Chapuy. A new combinatorial identity for unicellular maps, via a direct bijective approach. *Advances in Applied Mathematics*, 47(4) :874 – 893, 2011.
- [Che86] L. Paul Chew. There is a planar graph almost as good as the complete graph. In *2nd Annual ACM Symposium on Computational Geometry (SoCG)*, pages 169–177, August 1986.
- [Che89] L. Paul Chew. There are planar graphs almost as good as the complete graph. *Journal of Computer and System Sciences*, 39(2) :205–219, 1989.
- [Chu06] Julia Chuzhoy. Covering problems with hard capacities. *SIAM J. Comput.*, 36(2) :498–515, 2006.
- [CJSJ97] Robert Cori, Benjamin Jacquard, Gilles Schaeffer, and Robert. Cori Benjamin. Jacquard. Description trees for some families of planar maps. In *Formal Power Series and Algebraic Combinatorics, Proceedings of the 9th Conference, Vienna*, page 196–208, 1997.
- [CK97] M. Chrobak and G. Kant. Convex grid drawings of 3-connected planar graphs. *Internatl. Journal of Computational Geometry and Applications*, 7(3) :211–223, 1997.
- [Cla87] K. Clarkson. Approximation algorithms for shortest path motion planning. In *STOC '87 : Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 56–65. ACM, 1987.
- [CW99] Fabián A. Chudak and David P. Williamson. Improved approximation algorithms for capacitated facility location problems. In *IPCO*, pages 99–113, 1999.
- [DBTT92] G. Di Battista, R. Tamassia, and I. G. Tollis. Area requirement and symmetry display of planar upward drawings. *Discrete Comput. Geom.*, 7(4) :381–401, 1992.
- [DCKM04a] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi : a decentralized network coordinate system. *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 15–26, 2004.
- [dFOdMR95] H. de Fraysseix, P. Ossona de Mendez, and P. Rosenstiehl. Bipolar orientations revisited. *Discrete Appl. Math.*, 56(2-3) :157–179, 1995.
- [DFS90] David P. Dobkin, Steven J. Friedman, and Kenneth J. Supowit. Delaunay graphs are almost as good as complete graphs. *Discrete & Computational Geometry*, 5(4) :399–407, December 1990.

- [DG96] S. Dulucq and O. Guibert. Stack words, standard tableaux and Baxter permutations. *Discrete Math.*, 157(1-3) :91–106, 1996.
- [DGW96] S. Dulucq, S. Gire, and J. West. Permutations with forbidden subsequences and nonseparable planar maps. *Discrete Math.*, 153(1-3) :85–103, 1996.
- [Dil90] Michael B. Dillencourt. Toughness and delaunay triangulations. *Discrete and Computational Geometry*, 5(1) :575–601, December 1990.
- [dIVL81] W. Fernandez de la Vega and G.S. Lueker. Bin packing can be solved within $1 + \varepsilon$ in linear time. *Combinatorica*, pages 1 :349–355, 1981.
- [DR12] Mirela Damian and Kristin Raudonis. Yao graphs stan theta graphs. *Discrete Mathematics, Algorithms and Applications*, 04(02) :1250024, 2012.
- [EL08] L. Epstein and A. Levin. On Bin Packing with Conflicts. *SIAM Journal on Optimization*, 19(3) :1270–1298, 2008.
- [Far48] I. Fary. On straight lines representation of planar graphs. *Acta Sci. Math. Szeged*, 11 :229–233, 1948.
- [Fel01] S. Felsner. Convex drawings of planar graphs and the order dimension of 3-polytopes. *Order*, 18 :19–37, 2001.
- [Fel04a] S. Felsner. *Geometric Graphs and Arrangements*. Vieweg Verlag, 2004.
- [Fel04b] S. Felsner. Lattice structures from planar graphs. *Electron. J. Comb*, 11(1), 2004.
- [FFNO] S. Felsner, É. Fusy, M. Noy, and D. Orden. Bijections for Baxter families and related objects. ArXiv :0803.1546v1.
- [FPP88] H. de Fraysseix, J. Pach, and J. Pollack. Small sets supporting fary embeddings of planar graphs. In *20th Annual ACM Symp. on Theory of Computing*, pages 426–433, 1988.
- [FPP90] H. de Fraysseix, J. Pach, and J. Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10 :41–51, 1990.
- [FPS09] É. Fusy, D. Poulalhon, and G. Schaeffer. Bijective counting of plane bipolar orientations and Schnyder woods. *Europ. J. Combinatorics*, 30(7) :1646–1658, 2009. ArXiv :0803.0400v1.
- [FS09] P. Flajolet and R. Sedgewick. *Analytic combinatorics*. Cambridge University Press, Cambridge, 2009.
- [FSP08] Éric Fusy, Gilles Schaeffer, and Dominique Poulalhon. Dissections, orientations, and trees with applications to optimal mesh encoding and random sampling. *ACM Trans. Algorithms*, 4 :19 :1–19 :48, May 2008.
- [Fus06] É Fusy. Transversal structures on triangulations : combinatorial study and straight-line drawing. In *Proceedings of Graph Drawing'05*, volume 3843 of *Lecture Notes in Computer Science*, pages 177–188, 2006.
- [Fus10] Eric Fusy. Bijective counting of involutive baxter permutations. In *Bijective counting of involutive Baxter permutations*, 2010.
- [FZ08] Stefan Felsner and Florian Zickfeld. Schnyder woods and orthogonal surfaces. *Discrete and Computational Geometry*, 40(1) :103–126, July 2008.
- [Gir93] S. Gire. *Arbres, permutations à motifs exclus et cartes planaires : quelques problèmes algorithmiques et combinatoires*. PhD thesis, Université Bordeaux 1, 1993.
- [GL00] O. Guibert and S. Linusson. Doubly alternating Baxter permutations are Catalan. *Discrete Math.*, 217(1-3) :157–166, 2000.
- [GL12] Daniel Gonçalves and Benjamin Lévêque. Toroidal maps : Schnyder woods, orthogonal surfaces and straight-line representations. ArXiv :1202.0911, 2012.
- [GLM⁺04] Eric Goles, Matthieu Latapy, Clément Magnien, Michel Morvan, and Ha Duong Phan. Sandpile models and lattices : a comprehensive survey. *Theoretical Computer Science*, 322(2) :383 – 407, 2004.

- [GO97] Jacob E. Goodman and Joseph O'Rourke, editors. *Handbook of discrete and computational geometry*. CRC Press, Inc., Boca Raton, FL, USA, 1997.
- [GS00] D. D. Gebhard and B. E. Sagan. Sinks in acyclic orientations of graphs. *J. Combin. Theory Ser. B*, 80(1) :130–146, 2000.
- [GSG02] P. K. Gummadi, S. Saroiu, and S. D. Gribble. King : estimating latency between arbitrary internet end hosts. *Computer Communication Review*, 32(3) :11, 2002.
- [GZ83] C. Greene and T. Zaslavsky. On the interpretation of Whitney numbers through arrangements of hyperplanes, zonotopes, non-Radon partitions, and orientations of graphs. *Trans. Amer. Math. Soc.*, 280(1) :97–126, 1983.
- [HMS00] Tetsuya Hiroshima, Yuichiro Miyamoto, and Kokichi Sugihara. Another proof of polynomial-time recognizability of delaunay graphs. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E83-A(4) :627–638, April 2000.
- [Kan96] G. Kant. Drawing planar graphs using the canonical ordering. *Algorithmica*, 16 :4–32, 1996.
- [Kei88] J. M. Keil. Approximating the complete euclidean graph. In *No. 318 on SWAT 88 : 1st Scandinavian workshop on algorithm theory*, pages 208–213, London, UK, 1988. Springer-Verlag.
- [KG92] J. Mark Keil and Carl A. Gutwin. Classes of graphs which approximate the complete euclidean graph. *Discrete & Computational Geometry*, 7(1) :13–28, December 1992.
- [KH97] Goos Kant and Xin He. Regular edge labeling of 4-connected plane graphs and its applications in graph drawing problems. *Theoretical Computer Science*, 172(1–2) :175–193, 1997.
- [KLL09] M.J. Kao, C.S. Liao, and D.T. Lee. Capacitated domination problem. *Algorithmica*, pages 1–27, 2009.
- [KM07] Fabian Kuhn and Thomas Moscibroda. Distributed approximation of capacitated dominating sets. In *SPAA '07 : Proceedings of the nineteenth annual ACM symposium on Parallel algorithms and architectures*, pages 161–170, New York, NY, USA, 2007. ACM.
- [KP08] Iyad A. Kanj and Ljubomir Perković. On geometric spanners of Euclidean and unit disk graphs. In *25th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume hal-00231084, pages 409–420. HAL, February 2008.
- [KPR00] M.R. Korupolu, C.G. Plaxton, and R. Rajaraman. Analysis of a Local Search Heuristic for Facility Location Problems. *Journal of Algorithms*, 37 :146188, 2000.
- [KX12] Iyad A. Kanj and Ge Xia. Improved local algorithms for spanner construction. *Theor. Comput. Sci.*, 453 :54–64, September 2012.
- [Lar10a] H. Larchevêque. *Agrégation de ressources avec contrainte de distance*. PhD thesis, LaBRI, Université Bordeaux I, Talence, France, 2010.
- [LBSB09b] Cristian Lumezanu, Randy Baden, Neil Spring, and Bobby Bhattacharjee. Triangle inequality variations in the internet. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference, IMC '09*, pages 177–183, New York, NY, USA, 2009. ACM.
- [LEC67] A. Lempel, S. Even, and I. Cederbaum. An algorithm for planarity testing of graphs. In *Theory of Graphs (Internat. Sympos., Rome, 1966)*, pages 215–232. Gordon and Breach, New York, 1967.
- [LFV08] E. Lebar, P. Fraigniaud, and L. Viennot. The inframetric model for the internet. In *Proceedings of the 27th IEEE International Conference on Computer Communications (INFOCOM)*, pages 1085–1093, Phoenix, 2008.
- [LLS04] Ching-Chi Lin, Hsueh-I Lu, and I-Fan Sun. Improved compact visibility representation of planar graph via Schnyder's realizer. *SIAM Journal on Discrete Mathematics*, 18(1) :19–29, 2004.

- [LM02] Matthieu Latapy and Clémence Magnien. Coding distributive lattices with edge firing games. *Information Processing Letters*, 83(3) :125–128, 2002.
- [LSW04] Xiang-Yang Li, Ivan Stojmenovic, and Yu Wang. Partial delaunay triangulation and degree limited localized bluetooth scatternet formation. *IEEE Trans. Parallel Distrib. Syst.*, 15(4) :350–361, 2004.
- [Lub85a] M Luby. A simple parallel algorithm for the maximal independent set problem. In *Proceedings of the seventeenth annual ACM symposium on Theory of computing*, STOC '85, pages 1–10, New York, NY, USA, 1985. ACM.
- [LW04] Xiang-Yang Li and Yu Wang. Efficient construction of low weight bounded degree planar spanner. *International Journal of Computational Geometry and Applications*, 14(1-2) :69–84, 2004.
- [Mal79] C. L. Mallows. Baxter permutations rise again. *J. Combin. Theory Ser. A*, 27(3) :394–396, 1979.
- [Mil02] Ezra Miller. Planar graphs as minimal resolutions of trivariate monomial ideals. *Documenta Mathematica*, 7 :43–90, 2002.
- [NS07] Giri Narasimhan and Michiel Smid. *Geometric Spanner Networks*. Cambridge University Press, 2007.
- [OdM94] P. Ossona de Mendez. *Orientations bipolaires*. PhD thesis, École des Hautes Études en Sciences Sociales, 1994.
- [Pro93] J. Propp. Lattice structure for orientations of graphs. Manuscript, 1993.
- [PS03] Dominique Poulalhon and Gilles Schaeffer. Optimal coding and sampling of triangulations. In *30th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 2719 of LNCS, pages 1080–1094. Springer, July 2003.
- [PTW01] M. Pál, É. Tardos, and T. Wexler. Facility location with nonuniform hard capacities. In *FOCS '01 : Proceedings of the 42nd IEEE symposium on Foundations of Computer Science*, page 329, Washington, DC, USA, 2001. IEEE Computer Society.
- [RMK⁺09a] V. Ramasubramanian, D. Malkhi, F. Kuhn, M. Balakrishnan, A. Gupta, and A. Akella. On the treeness of internet latency and bandwidth. In *SIGMETRICS '09 : Proceedings of the eleventh international joint conference on Measurement and modeling of computer systems*, pages 61–72, New York, NY, USA, 2009. ACM.
- [Rot05] Günter Rote. Strictly convex drawings of planar graphs. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '05, pages 728–734, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.
- [RS91] Jim Ruppert and Raimund Seidel. Approximating the d -dimensional complete Euclidean graph. In *3rd Canadian Conference on Computational Geometry (CCCG)*, pages 207–210, 1991.
- [RT86] P. Rosenstiehl and R.E. Tarjan. Rectilinear planar layouts and bipolar orientations of planar graphs. *Discrete Comput. Geom.*, 1 :343–353, 1986.
- [Sch89] Walter Schnyder. Planar graphs and poset dimension. *Order*, 5 :323–343, 1989.
- [Sch90] Walter Schnyder. Embedding planar graphs on the grid. In *1st Symposium on Discrete Algorithms (SODA)*, pages 138–148. ACM-SIAM, January 1990.
- [Sch99] Gilles Schaeffer. Random sampling of large planar maps and convex polyhedra. In *31st Annual ACM Symposium on Theory of Computing (STOC)*, pages 760–769, Atlanta, Georgia, May 1999.
- [STA97] D.B. Shmoys, E. Tardos, and K. Aardal. Approximation algorithms for facility location problems. In *In Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, 1997.
- [Tho80] C. Thomassen. Planarity and duality of finite and infinite planar graphs. *Journal of Combinatorial Theory (B)*, 29 :244–271, 1980.

- [Tut60] W. T. Tutte. Convex representations of graphs. *Proc. London Math. Soc.*, 10 :304–320, 1960.
- [Ull71] J.D. Ullman. The performance of a memory allocation algorithm. Technical report 100, Princeton University, Princeton, NJ, 1971.
- [Vie84] G. Viennot. Bijective aspects of Baxter permutations. *Abstracts Amer. Math. Soc.*, 1984.
- [Wes95] J. West. Generating trees and the Catalan and Schröder numbers. *Discrete Math.*, 146(1-3) :247–262, 1995.
- [Xia11] Ge Xia. Improved upper bound on the stretch factor of delaunay triangulations. In *27th Annual ACM Symposium on Computational Geometry (SoCG)*, pages 264–273, June 2011.
- [XZ11] Ge Xia and Liang Zhang. Toward the tight bound of the stretch factor of Delaunay triangulations. In *23rd Canadian Conference on Computational Geometry (CCCG)*, August 2011.
- [Yao82] Andrew Chi-Chih Yao. On constructing minimum spanning trees in k -dimensional spaces and related problems. *SIAM Journal on Computing*, 11(4) :721–736, 1982.
- [ZH03] Huaming Zhang and Xin He. Compact visibility representation and straight-line grid embedding of plane graphs. In *8th International Workshop on Algorithms and Data Structures (WADS)*, volume 2748 of Lecture Notes in Computer Science, pages 493–504. Springer, July 2003.

Index

- A_i , 36
- C^p , 36
- $Comp(I, d)$, 55
- H_6 , 45
- H_9 , 41
- H_{S3} , 45
- M^+ , 29
- $P_i(v)$, 19
- T_i , 19
- $T_i(p, q)$, 39
- $V_T(p)$, 38
- \mathcal{A}_k , 36
- ℓ_C , 36
- \hat{M} , 6, 12
- \tilde{M} , 6
- \mathcal{E} , 34
- $parent_i(p)$, 36
- ρ -inframétrieque, 55
- $SUBTREE_{d_{\max}}(e)$, 61
- $charge(C)$, 42
- $children_i(u)$, 41
- $closest_i(u)$, 41
- $first_i(u)$, 41
- $last_i(u)$, 41
- i -distant, 42
- i -pertinent, 41
- w_{lost} , 59
- \mathcal{K}_1 , 58
- \mathcal{M}_1 , 58
- $mir(O)$, 6
- OPT_{BCCD} , 55
- $OPT_{BCCD}(\mathcal{I})$, 55
- $OPT_{BPCD}(\mathcal{I})$, 55
- étirement, 34
- 3-dégénéré, 14
- 3-orientation, 13, 22

- Θ_k -graphe orienté, 36

- angle, 23
- apex, 36
- arc orthogonal, 37

- BCCD, 55
- Bin covering avec contrainte de distance, 55

- Bin packing avec conflit, 55
- Bin Packing avec conflits, 55
- Bin packing avec contrainte de distance, 55
- bipolaire, 6
- bois de Schnyder, 13, 19
- bord droit, 6
- BPaC, 55
- BPCD, 55

- cône, 36
- cône négatif, 36
- cône positif, 36
- carte, 6
- carte enracinée, 6
- carte planaire, 6
- cellule de TD-Voronoi, 38
- charge, 41
- chemin canonique, 43
- chemin de Dyck, 13
- children(x), 61
- composition parallèle, 12
- composition série, 12
- coude de type i , 37
- coude géodésique, 37
- couvre, 5

- degré d'une face, 6
- degré sortant droit, 6
- degré sortant gauche, 6
- demi- Θ_k -graphe, 36
- descente, 4, 5
- dessin convexe, 18
- diagramme, 5
- diagramme de Hasse, 5
- diagramme de TD-Voronoi, 38
- diamètre d'une lentille, 60
- distance triangulaire, 38
- division, 23
- division d'une arête double, 23
- division-ccw, 23
- division-cw, 23
- division-cw courte, 24
- dualité, 6

- edge firing game, 14

- face, 6
- face droite, 6
- face externe, 6
- face interne, 6
- filtre, 36
- First-Fit-Decreasing, 61
- fusion, 23
- fusion du genou, 23

- genou, 23
- genou-ccw, 23
- genou-cw, 23
- Graphe de Compatibilité, 55

- homothétie, 38

- internement 3-connexe, 19, 29
- internement convexe, 30
- intervalle, 13

- last-mile, 65
- Lemme d'extension, 12
- Lentille de e_i et e_j , 60
- LOP, 11
- lr-maximum, 5
- lr-minimum, 5

- métrique d'arbre, 55
- métrique Vivaldi, 55
- maximum à droite, 4
- maximum à gauche, 4, 5
- montée, 4, 5, 7
- motifs barrés, 5

- Next-Fit-Decreasing, 58
- Next-Fit-Distance-Decreasing, 61
- noeud gardé, 65

- occurrence, 5
- orientation bipolaire plane, 6

- pôle, 6
- parent(x), 61
- permutation, 5
- permutation de Baxter, 4
- pièce orientée à droite, 11
- pièce orientée à gauche, 11
- piconet, 34
- plongement géodésique, 37
- poids étendu, 61
- puits, 6

- rl-maximum, 5
- rl-minimum, 5
- ROP, 11

- séparable, 6
- série-parallèle, 10, 12
- scatternett, 34
- semimétrique, 55
- simplement orienté, 37
- sommet droit, 6
- sommet gauche, 6
- sommets de suspension, 19
- source, 6
- spanner, 34
- stack triangulation, 14
- structures décomposables, 10
- structures transverses, 47
- surface orthogonale coplanaire, 36
- suspension, 19
- suspension complétée, 21
- suspension duale, 21

- TD-Delaunay, 38
- treillis de Kreweras, 13
- treillis de Stanley, 13
- treillis de Tamari, 13
- type d'arête, 25

- zone étendue, 58