Généralisation du diagramme de Voronoï et placement de formes géométriques complexes dans un nuage de points.

Thomas Iwaszko

▶ **To cite this version:**

Thomas Iwaszko. Généralisation du diagramme de Voronoï et placement de formes géométriques complexes dans un nuage de points.. Autre [cs.OH]. Université de Haute Alsace - Mulhouse, 2012. Français. NNT : 2012MULH4083 . tel-01005212

HAL Id: tel-01005212

https://theses.hal.science/tel-01005212v1

Submitted on 12 Jun 2014

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

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

université de haute alsace
école doctorale Jean-Henri Lambert ; ed 494

Thèse

présentée en vue d'obtenir le grade de Docteur,
spécialité informatique

par

Thomas Iwaszko

Généralisation du diagramme de Voronoï et placement de formes géométriques complexes dans un nuage de points

Thèse soutenue le 22 novembre 2012 devant le jury composé de :

MM.	Abderrahim Elmoataz	Université de Caen Basse-Normandie, GREYC	(examinateur)
	Lhassane Idoumghar	Université de Haute Alsace, LMIA	(codirecteur)
	Mahmoud Melkemi	Université de Haute Alsace, LMIA	(codirecteur)
	Dominique Michelucci	Université de Bourgogne, Le2i	(rapporteur)
	Ahmed Mostefaoui	Université de Franche-Comté, LIFC	(examinateur)
	René Schott	Université de Lorraine, Loria	(rapporteur)

« A mathematician, like a painter or poet, is a maker of patterns.
If his patterns are more permanent than theirs,
it is because they are made with ideas »

G.H.Hardy

Remerciements

Je tiens tout d'abord à exprimer ma reconnaissance à Mahmoud Melkemi et Lhassane Idoumghar, qui m'ont offert la chance de poursuivre un travail de thèse en informatique, sous leur direction. Cette expérience m'a appris énormément de choses et cela n'aurait pas été possible sans la patience, la confiance et l'ambition de mes encadrants.

J'exprime toute ma gratitude aux chercheurs qui ont bien voulu prendre le rôle de rapporteur de mon mémoire : Dominique Michelucci et René Schott, ainsi qu'aux chercheurs qui ont rejoint le jury : Abderrahim Elmoataz et Ahmed Mostefaoui.

Merci à Julien Chiquet, maître de conférences à Évry pour avoir mis à disposition sa classe LaTeX pour la rédaction de mémoires de thèse, celle-là même que j'ai choisi pour rédiger le présent document.

Merci à tous mes amis proches : Wael (Kinzeron), Willy, Nicole, Mickaël, Julien (Yul), Julien (Skill), Gauthier (Walter) et Fabien pour leur joie de vivre, leur sympathie, les promenades méditatives et autres expériences partagées qui m'ont tant inspiré et enrichi.

Merci à tous les gens de ma famille : Dorota, Piotr, Ida, Alicia, Sabine, Anna, Yvon *et al.* qui par leur présence et leur bienveillance m'ont apporté un précieux soutien.

Je remercie de tout cœur Ilona pour son amour, sa générosité et le courage dont elle a fait preuve à mes côtés.

Au cours de ces dernières années, il m'est arrivé de tomber sur divers textes que je juge -indépendamment du domaine- exceptionnels et grandioses.

Pour conclure cette page, j'aimerais rendre hommage aux quelques auteurs dont la lecture m'a impressionné à jamais et m'a permis de dépasser mes limites : Isaac Asimov, Philip K. Dick, Bernard Werber, Alan Turing, Alexis Carrel, Jean-Paul Delahaye, Dale Carnegie, Jean-Philippe Rennard, Epictète, Gilles Dowek, Jesse Schell, Matthieu Ricard. Je n'aurais certainement pas suivi ma curiosité aussi longtemps sans leur subtile influence, sans la grandeur de ces hommes qui ont bien voulu communiquer leurs idées à l'Humanité entière.

Strasbourg, le 4 septembre 2012.

Table des matières

Table des matières **vi**

Liste des figures **viii**

Liste des publications **1**

1 Présentation synthétique du travail réalisé **3**
1.1 Nouvelle généralisation du diagramme de Voronoï . . 5
1.1.1 Motivations pour la définition d'un nouveau diagramme 7
1.1.2 Aperçu du procédé de généralisation utilisé 7
1.1.3 Calcul du diagramme de Voronoï Paramétré par une Union de Disques 8
1.2 Placement de formes géométriques en présence d'obstacles 13
1.2.1 Motivations pour l'étude du placement 13
1.2.2 Travaux existants 14
1.2.3 Placement d'union de disques dans un nuage de points . 15
1.2.4 Résolution générique de problèmes de placement 15

2 Modèle à base de disques pour les formes géométriques planes **19**
2.1 Représentation de formes géométriques à base de disques 21
2.2 Notations et définitions 24
2.2.1 Ensemble et union de disques standard 24
2.2.2 Notion de configuration 26
2.2.3 Copie paramétrée et fonction-centre 27
2.2.4 Copie vide 29
Conclusion du chapitre 29

I Diagramme de Voronoï et généralisation basée sur une union de disques **31**

3 Nouvelle généralisation du diagramme de Voronoï **33**
3.1 Rappels sur les régions de Voronoï 35
3.2 Généralisations du diagramme de Voronoï 35
3.2.1 Motivations 35
3.2.2 Généralisations existantes 36
3.3 Principe de la nouvelle généralisation 39
3.3.1 Prédicat du disque vide 39
3.3.2 Extension du prédicat aux unions de disques 41

3.3.3 Régions de Voronoï Paramétrées par une Union de Disques 41
Conclusion du chapitre 48

4 Construction de régions de Voronoï paramétrées par une union de disques **49**
4.1 Caractérisation algébrique d'une région de VPUD . . . 51
4.2 Frontière de la région de VPUD 55
4.2.1 Rappels sur l'étude de courbes planes 55
4.2.2 Propriétés de la frontière d'une région de VPUD 56
4.2.3 Équation caractéristique de la frontière de sur-région élémentaire 57
4.2.4 Inclusion de la courbe-frontière dans une quartique . . . 58
4.3 Propriétés globales des régions de VPUD 61
4.3.1 Inclusion de régions de VPUD aux paramètres similaires 61
4.3.2 Angles interdits 62
4.3.3 Non partitionnement du plan par régions de VPUD . . . 64
4.3.4 Non connexité d'une région de VPUD 65
4.4 Algorithmes calculant le diagramme de VPUD 65
4.4.1 Difficultés inhérentes à la représentation sur machine . . 67
4.4.2 Approximation discrète basée sur des points isolés 68
4.4.3 Approximation intérieure basée sur des pavés 79
Conclusion du chapitre 85

II Placement de formes géométriques en présence d'obstacles 89

5 Placement d'union de disques dans un nuage de points **91**
5.1 Travaux existants en placement de formes géométriques 93
5.1.1 Motivations pour l'étude du placement de formes géométriques 94
5.1.2 Comparaison des problèmes de placement 97
5.1.3 Placement et packing 102
5.2 Placement d'union de disques : définition et notations 103
5.3 Espace de configurations 104
5.4 Problèmes de placement traités dans la thèse 106
5.4.1 Placement par rotation uniquement 106
5.4.2 Placement par homothétie uniquement 107
5.4.3 Placement par homothétie et rotation combinées 109
5.4.4 Placement par translation 112
5.4.5 Cas particulier d'un espace de configuration fini 112
Conclusion du chapitre 113

6 Algorithme générique de placement de formes géométriques complexes **115**
6.1 Résolution du placement par rotation 117
6.1.1 Rappels et identification d'un cas particulier 117
6.1.2 Définition d'orbite et d'arc interdit 118
6.1.3 Calcul de l'ensemble des configurations autorisées 120
6.2 Méthode générique de résolution des pp inversibles . . 125
6.2.1 Problèmes de placement inversibles 125

6.2.2 Généralisation des notions d'orbite et arc interdit 126
6.2.3 Résolution générique de tout PP inversible 128
6.2.4 Compactage des résultats 133
6.3 Exemples d'application de la méthode 135
6.3.1 Placement par homothétie uniquement 135
6.3.2 Placement par homothétie et rotation combinées 137
6.3.3 Placement par translation uniquement 140
Conclusion du chapitre 145

Conclusion et perspectives **147**

A Annexes **149**
A.1 Preuve de la finitude des points de jonction 151
A.2 Existence de cas particuliers simplifiant la frontière de sur-régions élémentaires 151
A.3 Preuve de la nature quartique des courbes type $\mathcal{C}'$ . . . 155
A.4 Preuve de la non-connexité d'une région de VPUD . . . 156
A.5 Implémentation de la résolution du placement par homothétie 159
A.6 Justification de l'interprétation des zones interdites, dans le contexte du PP par homothétie-rotation combinées 162
A.7 Perspectives quant au calcul exact d'une région de VPUD 165

Bibliographie **177**

Notations **183**

Liste des figures

1.1 Représentation de formes géométriques complexes par union de disques 4
1.2 Diagramme de Voronoï pour un ensemble arbitraire de vingt-sept points 5
1.3 Applications concrètes du diagramme de Voronoï 6
1.4 Prédicat du disque vide et région de Voronoï 9
1.5 Copie d'une union de disques 10
1.6 Copies considérées par le prédicat d'union de disques vide 11
1.7 Prédicat d'union de disques vide et région de Voronoï généralisée 12
1.8 Placement d'une union de disques dans un nuage de points par rotation 17
1.9 Exemple de solution au placement par rotation 18

2.1 Exemples de représentation par axe médian d'une forme géométrique définie par son contour. Auteurs des illustrations : QUB -Finite Element Research Group 2012, http ://sog1.me.qub.ac.uk/ 22
2.2 Méthode possible pour passer d'une représentation polygonale à union de disques, auteur de l'image : Tam (2004) . . . 23
2.3 Ensemble de Disques Standard et union de disques 25
2.4 Représentation de formes géométriques complexes par union de disques 25
2.5 Notion de copies d'une union de disques. Pour chaque copie représentée, la position du premier centre et la valeur du premier rayon sont indiquées 28

3.1 Généralisation du diagramme de Voronoï à l'espace 3D . . . 37
3.2 Généralisation du diagramme de Voronoï aux segments . . 38
3.3 Généralisation du diagramme de Voronoï à la métrique L_1, aussi connue sous le nom de distance de Manhattan 38
3.4 Généralisation du diagramme de Voronoï à une divergence de Bregman arbitrairement choisie. 38
3.5 Région de Voronoï classique et prédicat de disques vide . . 40
3.6 Prédicat d'union de disques vide 42
3.7 Copies considérées par le prédicat d'union de disques vide 43
3.8 Régions de Voronoï Paramétrée par une Union de Disques (exemple 1). À chaque germe est associée une région de couleur différente 45
3.9 Régions de Voronoï Paramétrée par une Union de Disques (exemple 2). À chaque germe est associée une région de couleur différente 46
3.10 Régions de Voronoï Paramétrée par une Union de Disques (exemple 3). À chaque germe est associée une région de couleur différente 47

4.1 Ensemble de Disques Standard (EDS) utilisé dans les figures suivantes 52
4.2 Décomposition d'une région de VPUD en sur-régions. L'intersection ensembliste des trois sur-régions représentées par (b,c,d) est égale à la région de VPUD représentée par (a). L'intérieur d'une région est représenté en gris 53
4.3 Décomposition d'une sur-région en sur-régions élémentaires. L'intersection ensembliste des deux sur-régions élémentaires représentées par (b,c) est égale à la sur-région représentée par (a). L'intérieur d'une région est représenté en gris 53
4.4 Exemples de courbes planes parmi les plus connues. 56
4.5 Non-linéarité de la frontière des sur-régions élémentaires . 57
4.6 Caractère non polygonal d'une région de VPUD. 60
4.7 Propriété d'inclusion entre régions de VPUD aux paramètres similaires 63
4.8 Exemple de non-connexité des régions de VPUD. La région $V_B(p_9)$ a la particularité d'être non-connexe 66

4.9 Résultats produits par l'algorithme 1 force brute (jeu de données #1) . . . 69
4.10 Résultats produits par l'algorithme 1 force brute (jeu de données #2) . . . 70
4.11 Résultats produits par l'algorithme 1 force brute (jeu de données #3) . . . 71
4.12 Résultats produits par l'heuristique 2 de visualisation rapide (jeu de données #1) . . . 76
4.13 Résultats produits par l'heuristique 2 de visualisation rapide (jeu de données #2) . . . 77
4.14 Résultats produits par l'heuristique 2 de visualisation rapide (jeu de données #3) . . . 78
4.15 Résultats de l'algorithme 3 (1) . . . 86
4.16 Résultat produit par l'algorithme 3 . . . 87

5.1 « Polygon containment problem » introduit par Chazelle . . 93
5.2 « Maximal Empty Rectangle problem » étudié par Chazelle et al. . . . 94
5.3 Application du placement de formes géométriques dans l'industrie textile (maroquinerie) . . . 95
5.4 Application du placement de formes géométriques en robotique : planification de trajectoires . . . 96
5.5 Détection de collision en 2D entre formes géométriques simples . . . 96
5.6 Différence de représentation des obstacles, entre problèmes décrits dans la littérature . . . 99
5.7 Différence de forme géométrique à placer, entre problèmes décrits dans la littérature . . . 100
5.8 Exemples de solutions au problème du « circle packing in a square » . . . 103
5.9 Problème du placement d'union de disques dans un nuage de points . . . 105
5.10 Copies considérées par le PP par rotation . . . 107
5.11 Placement d'union de disques par rotation . . . 108
5.12 Copies considérées par le PP par homothétie . . . 109
5.13 Problème de placement par homothétie . . . 110
5.14 Problème de placement par homothétie et rotation combinées 111
5.15 Copies considérées par le placement par translation . . . 112

6.1 Placement d'union de disques par rotation . . . 117
6.2 Cas particulier pour le PP par rotation : non vacuité du premier disque . . . 118
6.3 Orbite dans le contexte du placement par rotation . . . 119
6.4 Exemple d'arc interdit associé à un point de P et à la troisième orbite . . . 120
6.5 Fonction-centre réciproque dans le contexte du placement par rotation . . . 122
6.6 Solution calculée pour une instance du placement par rotation 124
6.7 Solution fournie par l'algorithme 6 pour un PP par homothétie 138

6.8 Solution fournie par l'algorithme 6 pour répondre au PP par homothétie-rotation combinées (1) 141
6.9 Solution fournie par l'algorithme 6 pour répondre au PP par homothétie-rotation combinées (2) 142
6.10 Solution fournie par l'algorithme 6 pour répondre au PP par homothétie-rotation combinées (3) 143
6.11 Solution fournie par l'algorithme 6 pour un PP par translation 146

A.1 Cas particuliers concernant la courbe délimitant une sur-région élémentaire 154
A.2 Exemple de région de VPUD non-connexe (en bleu) 159
A.3 Ensemble de disques utilisé pour la preuve de non-connexité 160
A.4 Unification des concepts présentés dans le mémoire. Le calcul d'une région de Voronoï classique est un cas particulier du calcul de région de VPUD. Ce calcul lui-même est un cas particulier de résolution du placement d'union de disques dans un nuage de points 168
A.5 Solutions au PP par translation d'un disque dans un nuage de trois points 171
A.6 Solutions au PP par translation-homothétie combinées d'un disque dans un nuage de trois points 171
A.7 Solutions au PP par translation d'une union de trois disques dans un nuage de trois points 172
A.8 Solutions au PP par translation-homothétie d'une union de trois disques dans un nuage de trois points 172
A.9 Obtention des solutions au PP spécial en mettant à profit le lien calculatoire 173
A.10 Projection orthogonale des solution au PP spécial sur le plan de coordonnée $z = 0$, noté $H(x, y, 0)$. Le résultat est égal à la frontière d'une sur-région élémentaire, région définie et étudiée au chapitre 4 174

Liste des publications

Article de journal en préparation

– *Shape placement in a set of points : finding and grouping solutions by using rotated copies*

Les travaux de recherche exposés dans ce mémoire ont donné lieu aux communications scientifiques listées ci-dessous.

Conférences internationales avec comité de lecture et actes de conférence

– *A Geometric Data Structure Applicable to Image Mining and Retrieval*
Proceedings of the International Conference on Image Analysis and Recognition, ICIAR 2010 (Póvoa de Varzim, Portugal), LNCS 6111, (2010), 364-373.
– *A Theoretical Structure for Computational Geometry : Regions of Point-free Overlapping Circles*
Proceedings of the International Conference on Signal Processing Computational Geometry and Artificial Vision, ISCGAV 2009 (Moscow, Russia), WSEAS Press, (2009), 117-122.
– *Regions of Empty Overlapping Circles*
Abstracts from the 7th Japan Conference on Computational Geometry and Graphs, JCCGG 2009 (Kanazawa, Japan), 81-82.

Conférence nationale avec actes de conférence

– *Diagrammes de Voronoï et placement de formes géométriques*
Actes des journées du Groupe de Travail en Modélisation Géométrique , GTMG'2011, Grenoble, mars 2011.

Conférence nationale

– *Variante de la décomposition de Voronoï paramétrée par une forme géométrique plane*
Journées de Géométrie Algorithmique (JGA'2010), CIRM, Marseille-Luminy, mars 2010.

Tous ces articles sont signés :
Thomas Iwaszko, Mahmoud Melkemi et Lhassane Idoumghar.

1 Présentation synthétique du travail réalisé

Le présent mémoire décrit les travaux de recherche menés au LMIA [1] de Mulhouse, dans le cadre d'une thèse avec allocation de recherche MENRT [2]. Dans ce premier chapitre, nous donnons une vue d'ensemble du travail réalisé et pour chaque sujet, nous faisons référence au chapitre dédié afin d'aider le lecteur à s'orienter dans la structure du document.

En *géométrie algorithmique*, on résout des problèmes de nature géométrique par le calcul. Pour y parvenir, il peut être nécessaire de définir ses propres outils : des modèles mathématiques et des structures de données spécifiques.

Avant d'aborder les problèmes étudiés au cours de nos travaux, le chapitre 2 introduit une représentation de formes géométriques complexes. Le choix est fait d'approximer de telles formes en accumulant plusieurs disques ayant leurs positions et tailles propres, comme c'est illustré par la figure 1.1. Une telle représentation offre de nombreux avantages :

- elle permet de modéliser des formes complexes, quelque soient leurs caractéristiques. La forme peut être non convexe, avec un ou plusieurs trous, *etc.* Il suffit d'employer davantage de disques pour avoir une approximation satisfaisante ;
- étant donnée une union de disques, une région similaire à celle-ci peut être désignée de manière pratique en considérant un nouveau centre et un nouveau rayon, de manière analogue au cercle ;
- le fait qu'un ensemble de disques soit décomposable en éléments indépendants facilite la conception d'algorithmes simples, traitant les disques de façon itérative, ou de manière parallèle ;
- le choix du disque comme « brique de base » produit des algorithmes efficaces, économiques en temps de calcul. Le caractère vide/non-vide d'un disque peut en effet être testé via un simple calcul de distance euclidienne et un test. Certains coûts algorithmiques peuvent aussi être réduits grâce au diagramme de Voronoï dont la définition utilise implicitement les propriétés du disque ;
- la construction d'un ensemble de disques approximant une forme géométrique a déjà été bien étudiée dans la littérature par Amenta et al. (2001), Tam (2004). Ce problème équivaut à discrétiser ce que l'on nomme en traitement d'images l'*axe médian* ou le *squelette topologique cf.* Blum (1967), Katz et Pizer (2001), Tam et Heidrich (2003) ;

1. Laboratoire de Mathématiques, Informatique et Applications
2. Ministère de l'Éducation Nationale, de la Recherche et de la Technologie

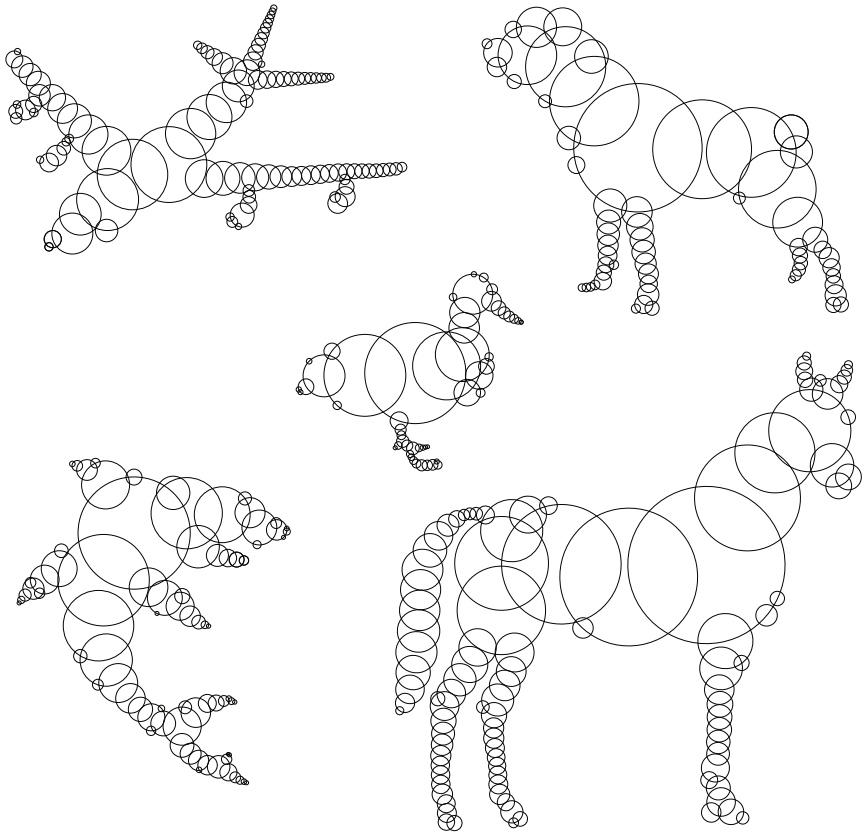


FIGURE 1.1 – *Représentation de formes géométriques complexes par union de disques*

– la modification de quelques disques (position, taille, existence du disque) composant une union de disques n'influence qu'en partie le résultat de calculs réalisés. Dans un tel cas, la sortie d'un algorithme pourrait être mise à jour au lieu d'être recalculée entièrement. Un tel procédé, en économisant des opérations de calcul, rendrait plus efficaces les algorithmes utilisant des données changeantes au cours du temps.

Ainsi, dans notre contexte, toute forme géométrique correspond à l'union ensembliste $\cup B$ des disques d'un ensemble fini $B = \{b_1, \ldots, b_m\}$. Un récapitulatif de toutes les notations mathématiques utilisées se trouve en fin d'ouvrage. Les éléments du chapitre 2 : la définition détaillée des unions de disques utilisées et le modèle pour manipuler celles-ci, constituent donc un prérequis à l'ensemble de nos travaux.

Le cœur du document : le texte allant du chapitre 3 au chapitre 6, se compose de deux parties distinctes dédiées aux objectifs suivants :

– proposer une nouvelle *généralisation du diagramme de Voronoï*, des moyens pour représenter et calculer sur ordinateur le résultat obtenu ;
– étudier différents problèmes de *placement d'union de disques dans un nuage de points* et proposer une *méthode générique* pour leur résolution.

Pour donner une idée intuitive de ce à quoi correspondent nos réalisations, nous détaillons ces objectifs l'un après l'autre, en décrivant leur contexte respectif, la démarche employée et un aperçu des résultats.

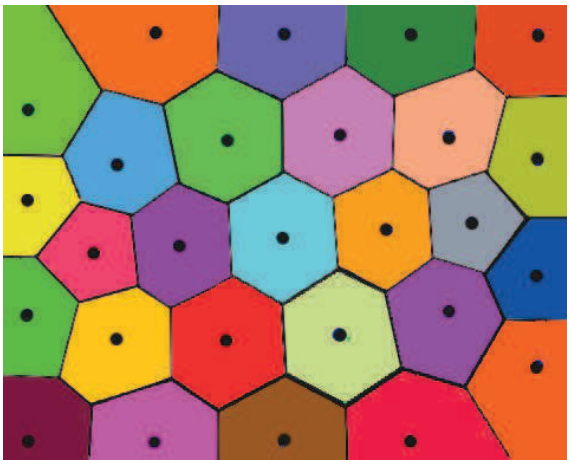


FIGURE 1.2 – *Diagramme de Voronoï pour un ensemble arbitraire de vingt-sept points*

1.1 NOUVELLE GÉNÉRALISATION DU DIAGRAMME DE VORONOÏ

S'il existe une structure de données emblématique de la géométrie algorithmique, c'est bien le *diagramme de Voronoï*. Celui-ci a été étudié très en détail ces trente dernières années, comme en témoigne l'état de l'art de Aurenhammer (1991).

Rappelons qu'un tel diagramme est défini pour un ensemble fini de points $P = \{p_1, \ldots, p_n\}$. Il consiste en une partition du plan en n régions, telles que chaque région $V(p_i)$ est égale à l'ensemble des points du plan plus proches de p_i que de $p_j, j \neq i$. Un exemple est représenté sur la figure 1.2.

Rappelons aussi que l'usage du diagramme de Voronoï s'est largement répandu en informatique depuis les années 1970 et a même dépassé ce contexte. Pour en donner un aperçu, voici quelques applications listées dans le livre de Okabe et al. (2000) :

- en astronomie, le diagramme de Voronoï est employé pour identifier des amas d'étoiles et de galaxies ;
- en sciences de l'environnement, les régions de Voronoï servent à modéliser la croissance et la compétition d'espèces végétales d'un écosystème (compétition pour occuper un espace vital permettant la croissance végétale) ;
- en reconnaissance de formes (*pattern recognition*), un descripteur de formes unidimensionnel simple portant le nom de « squelette » est extrait à partir de formes géométriques 2D, grâce à la recherche de contours et à un traitement basé sur le diagramme de Voronoï ;
- en robotique, le diagramme de Voronoï couplé à des notions de théorie des graphes fournit une solution simple et élégante à la planification de trajectoires (*motion planning*). La planification de trajectoires permet à un robot mobile de se déplacer tout en évitant des obstacles physiques présents dans l'environnement.

La figure 1.3 montre que le diagramme de Voronoï se retrouve jusque dans la nature et par mimétisme, dans le goût esthétique et les constructions humaines. Les applications diverses de ce diagramme expliquent l'intérêt que lui portent les chercheurs. Ces travaux existants sont décrits avec plus de précision en début de chapitre 3.

(a) *Modélisation dans le domaine biologique pour expliquer les motifs observés dans les ailes d'une libellule*

(b) *Conception en architecture pour choisir un design de construction original et esthétique*

FIGURE 1.3 – *Applications concrètes du diagramme de Voronoï*

1.1.1 Motivations pour la définition d'un nouveau diagramme

Au vu de ce que l'on vient de rappeler, *la généralisation du diagramme est intéressante en ceci qu'elle permet d'accroître davantage la portée applicative de cet outil.* Ce point de vue est défendu dans l'état de l'art de Aurenhammer et Klein (2000) où les auteurs listent les généralisations du diagramme de Voronoï et les *nouvelles* applications associées.

Par exemple : la généralisation du diagramme de Voronoï à la métrique L_1 ou L_∞ permet de résoudre le problème dit du « largest empty square problem » *cf.* l'article de Lee et Wong (1980). Des problèmes analogues qui considèrent une autre forme géométrique *simple* : carré, rectangle, ovale, *etc.* peuvent être résolus en considérant une métrique *ad-hoc*. En revanche, le placement de type *Largest Empty ...* (où les points de suspension sont à remplacer par une dénomination de forme géométrique plane) avec une forme à placer qui serait plus complexe : non-convexe, non-symétrique ou comportant un ou plusieurs trou(s) est un problème certainement hors de portée du diagramme de Voronoï classique. La méthode connue ne s'applique pas et ce problème reste, à notre connaissance, un problème ouvert.

Le diagramme de VPUD que nous présentons dans ce document permet justement de dépasser les limitations du diagramme de Voronoï et possède plusieurs avantages :

- il constitue un outil pertinent pour résoudre tout problème de placement du type *Largest Empty ...* pour des formes géométriques complexes, sans être limité par la complexité de la forme considérée ;
- il permet de grouper, de structurer en régions planes tout un ensemble de formes complexes vides ;
- il génère de nouvelles possibilités applicatives du fait qu'il étend plusieurs des propriétés du diagramme de Voronoï à des formes géométriques plus complexes. On peut envisager son application à différents problèmes de robotique où la forme du robot joue un rôle important et ne peut être approximée par un point ou un disque ;
- une partie des calculs nécessaires à son utilisation sur ordinateur se rapporte directement au calcul du diagramme de Voronoï classique. Ceci rend éventuellement possible la réutilisation ou la modification d'algorithmes existants pour calculer le nouveau diagramme.

1.1.2 Aperçu du procédé de généralisation utilisé

Au cours de nos travaux, nous avons généralisé le diagramme de Voronoï suivant une démarche originale dont nous donnons un aperçu ci-dessous. Les détails omis ici et la définition complète des régions de Voronoï généralisées sont donnés dans la *deuxième moitié* du chapitre 3.

1. On observe qu'une région de Voronoï est *aussi* égale à l'ensemble des points satisfaisant ce que nous appelons *prédicat du disque vide*. Celui-ci s'applique à un triplet de paramètres P, p_i, x avec P un ensemble fini de points, p_i un germe (*i.e.* un point membre de P) et x un point du plan arbitrairement choisi. Le prédicat s'énonce comme suit : l'intersection entre l'ensemble P et le disque ouvert centré en x, de rayon égal à la distance $\|x - p_i\|$ est vide. Ainsi, un point quel-

conque x fait partie de la région de Voronoï associée au point p_i, si et seulement si il vérifie le prédicat de disque vide. Formellement :

$$x \in V(p_i) \Leftrightarrow b(x, \|x - p_i\|) \cap P = \emptyset$$

Cette équivalence est illustrée à la figure 1.4.

2. Nous avons alors étendu ce prédicat à des formes plus complexes représentées par une union de disques [3] notée $\cup B$. Pour ce faire, on définit d'abord la notion de *copie* qui est obtenue à partir de $\cup B$ par combinaison de rotation, homothétie, translation. Nous notons toute copie $\cup B(x, \lambda, \theta)$ ou $\cup B(a)$ de manière plus compacte. Une copie dépend notamment, et comme le cercle, d'un centre x (renommé *point de référence*) et d'un rayon λ (renommé *taille de la copie*). La figure 1.5 donne un exemple de copie. Par suite, on note $\cup B(x, \|x - p_i\|, 0)$ une copie qui « touche » p_i ayant x comme point de référence, $\|x - p_i\|$ comme taille, mais qui garde l'orientation originelle de l'union $\cup B$. Cette invariance d'orientation est exprimée par le troisième paramètre, de valeur 0 qui correspond à l'angle de rotation. Une telle copie est illustrée via la figure 1.6. Avec cette notation, la région de Voronoï se généralise comme suit : un point x fait partie de la région de Voronoï généralisée associée à p_i, notée $V_B(p_i)$ si et seulement si il vérifie un prédicat plus général :

$$x \in V_B(p_i) \Leftrightarrow \cup B(x, \|x - p_i\|, 0) \cap P = \emptyset$$

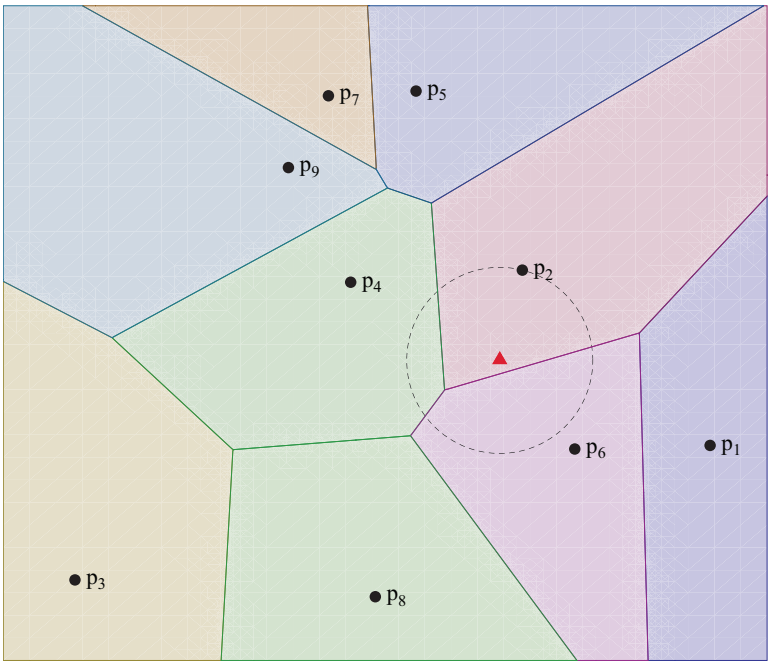
On appelle régions de Voronoï Paramétrées par Union de Disques (ou régions de VPUD) ces nouvelles régions généralisées. Celles-ci sont représentées par la figure 1.7. Le *prédicat d'union de disques vide* est bien l'extension à un cas plus général du prédicat de disque vide. Le nouveau prédicat s'applique cette fois à un quadruplet P, p_i, B, x au lieu d'un triplet P, p_i, x. Le paramètre B supplémentaire dénote l'ensemble de disques servant de « modèle » pour une forme géométrique complexe.

1.1.3 Calcul du diagramme de Voronoï Paramétré par une Union de Disques

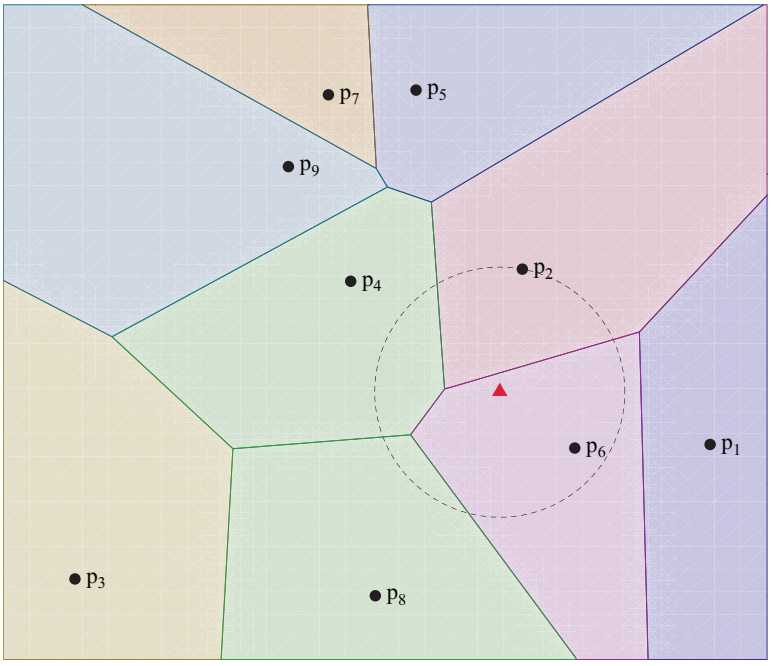
Au chapitre 4 nous exposerons de nombreuses propriétés concernant les régions de VPUD. Ces connaissances nous ont permis de déterminer une représentation sur machine pertinente et des algorithmes pour calculer le diagramme de VPUD. Ce dernier diagramme étant, à notre connaissance un concept original, *les algorithmes présentés sont les tout premiers algorithmes conçus pour réaliser un tel calcul*. Plus précisément, ce sont trois procédés différents que nous avons conçus pour approximer les régions de VPUD à une précision souhaitée et pour les afficher sur ordinateur.

Les deux premiers algorithmes en question utilisent une représentation point par point, ils peuvent produire des images de moyenne résolution. Le troisième utilise un ensemble de « pavés » *i.e.* d'intervalles bidimensionnels qui peuvent être de tailles différentes. On esquisse ici le principe du calcul pour chacun de ces trois algorithmes.

3. union de disques vérifiant quelques conditions, comme expliqué au chapitre 2.

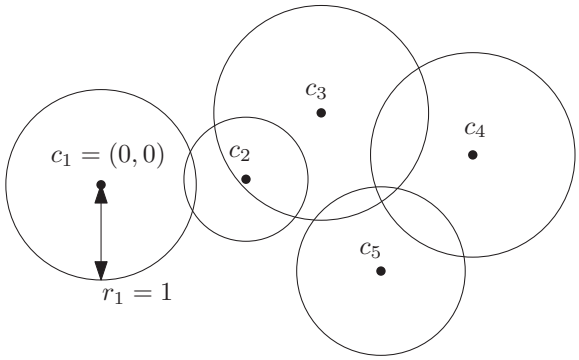


(a) *le point représenté par un triangle appartient bien à la région de Voronoï associée au germe p_2, car il satisfait le prédicat du disque vide*

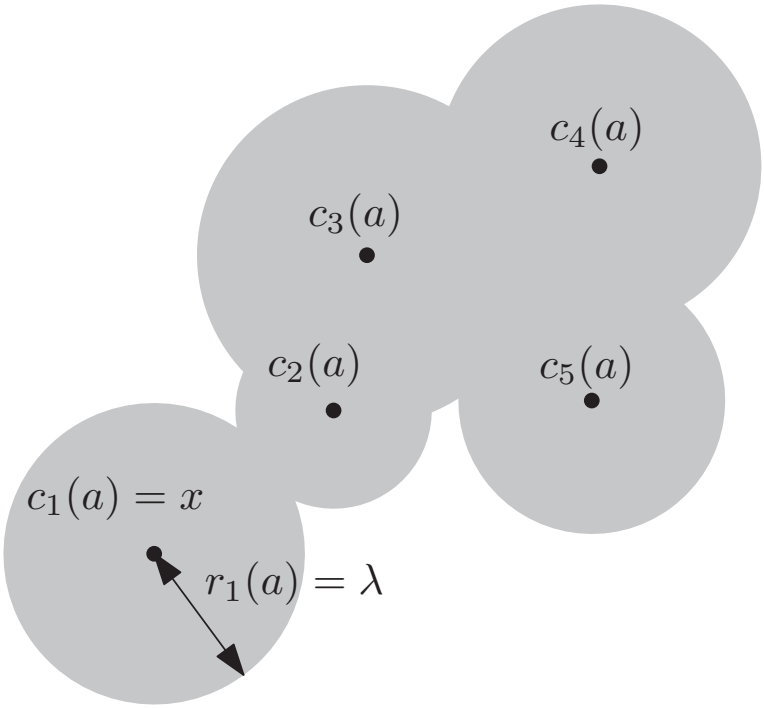


(b) *le point représenté par un triangle n'appartient pas à la région de Voronoï associée au germe p_2, car il ne satisfait pas le prédicat du disque vide*

FIGURE 1.4 – *Équivalence entre : appartenance d'un point à une région de Voronoï classique et vérification du prédicat de disque vide*

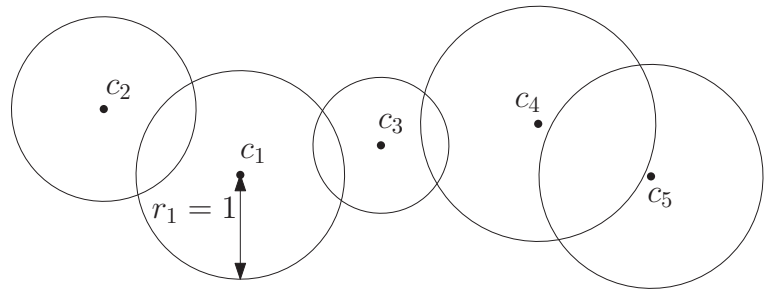


(a) *ensemble de disques, noté B, servant de modèle pour représenter une forme géométrique*

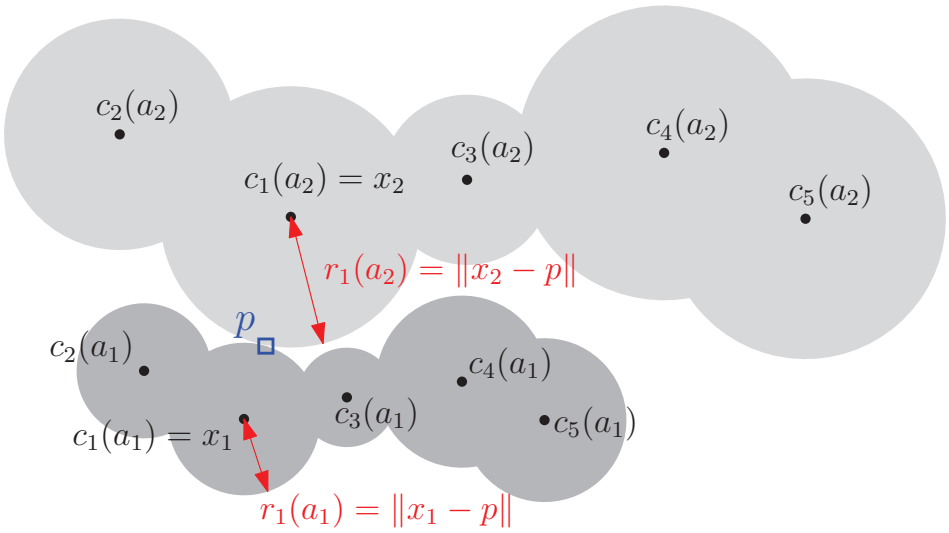


(b) *exemple de copie $\cup B(a)$ obtenue à partir de l'union $\cup B$. Le point x est appelé point de référence, le réel λ est la taille de la copie*

FIGURE 1.5 – *Copie d'une union de disques obtenue par application d'une combinaison de translation, homothétie, rotation sur l'union de disques originelle. Toute copie peut être identifiée par un triplet de paramètres $a = (x, \lambda, \theta)$*

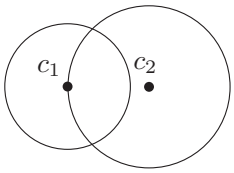


(a) *ensemble de disques B utilisé pour l'exemple*

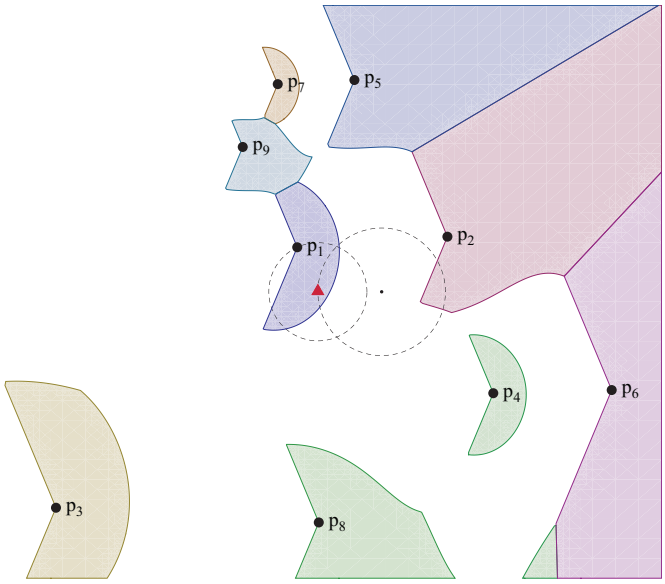


(b) *Deux copies* $\cup B(x_1, \|x_1 - p\|, 0)$ *et* $\cup B(x_2, \|x_1 - p\|, 0)$ *considérées par le prédicat d'union de disques vide*

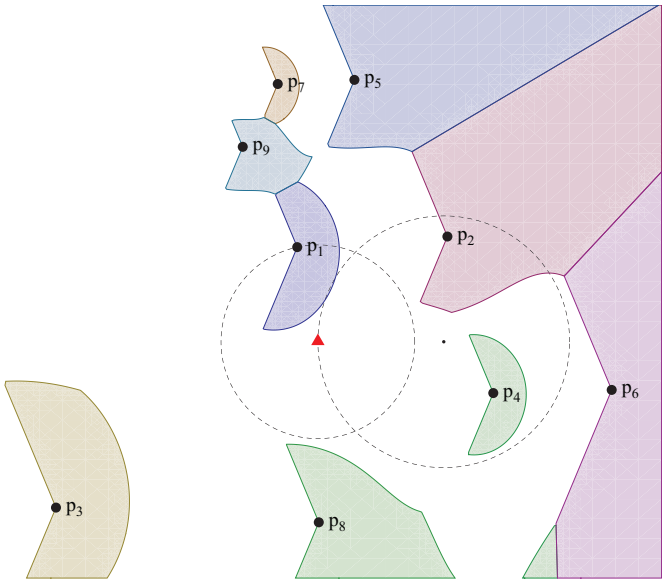
FIGURE 1.6 – *Toutes les copies considérées par le prédicat ont une même orientation et respectent la condition suivante : la taille de copie est égale à la distance entre le point de référence x choisi et un germe p donné (indiqué ici par un carré bleu)*



(a) *ensemble de disques utilisé pour représenter la forme*



(b) *la copie (représentée en pointillés) dont le point de référence x_1 est choisi comme montré par le triangle ci-dessus et dont la taille est $\|x_1 - p_1\|$ a une intersection vide avec le nuage de points P. Le prédicat d'union de disques vide est satisfait, donc le point x_1 est membre de la région de VPUD associée au germe p_1 (région représentée en bleu)*



(c) *la copie (représentée en pointillés) dont le point de référence x_2 est choisi comme montré par le triangle ci-dessus et dont la taille est $\|x_2 - p_1\|$ possède au moins un point de P en son intérieur. Le prédicat d'union de disques vide n'est pas satisfait, donc le point x_2 est exclu de la région de VPUD associée au germe p_1 (région représentée en bleu)*

FIGURE 1.7 – *Prédicat d'union de disques vide servant à définir les régions de Voronoï généralisées. On nomme régions de VPUD ces nouvelles régions*

Le premier algorithme consiste en une recherche par force brute. Le plan euclidien est borné par une boîte englobante et est discrétisé à une certaine résolution. Il en résulte une matrice de pixels. Connaissant la position des points de P, on teste chaque pixel individuellement pour vérifier si le prédicat d'union de disques vide est satisfait, c'est-à-dire si les coordonnées associées au pixel désignent un point d'une région de VPUD. Le résultat final est obtenu en associant une couleur par région de VPUD calculée.

Le deuxième algorithme correspond plus précisément à une heuristique probabiliste (procédé de calcul fournissant une réponse approchée et employant des nombres pseudo-aléatoires). Il réalise lui aussi une recherche pixel par pixel, mais celle-ci n'est ni exhaustive ni déterministe. L'heuristique se sert d'un découpage en « blocs » de pixels et ajuste dynamiquement les probabilités de tirage aléatoire dans chaque bloc pour améliorer son résultat. Les tests empiriques que nous avons réalisés montrent que cette heuristique a l'avantage de produire des images représentatives des régions de VPUD, en un temps bien plus court que l'algorithme « recherche par force brute ». Le résultat affiché est néanmoins sans garantie en ce qui concerne la topologie des régions.

Le troisième algorithme utilise quant à lui les techniques de l'*arithmétique des intervalles* pour tester l'appartenance de pavés aux régions de VPUD. Si le test d'un pavé ne donne pas suffisamment d'information, l'algorithme utilise une *bissection récursive* (le pavé courant est « coupé » en quatre sous-pavés) pour réduire pas à pas la taille des pavés testés et ainsi converger. L'algorithme a pour avantage de produire une approximation intérieure des régions de VPUD, avec une garantie mathématique sur la validité du résultat.

Ces trois procédés permettent d'employer les diagrammes de VPUD lors de calculs concrets sur ordinateur. L'utilisateur peut choisir l'un ou l'autre d'entre eux selon ses objectifs propres. Ils ont tous trois effectivement été implémentés via les langages de programmation C et Python. Le chapitre 4 présente le détail des algorithmes, de leurs résultats et une brève comparaison des performances en termes de durée de calcul.

1.2 Placement de formes géométriques en présence d'obstacles

Le chapitre 5 décrit une famille de problèmes récurrents en géométrie algorithmique. Ceux-ci concernent l'aménagement de l'espace, nous les désignons via un terme générique : « problèmes de placement ». Voyons le contexte avant de résumer nos contributions.

1.2.1 Motivations pour l'étude du placement

Les problèmes de placement sont couramment étudiés dans le but d'être appliqués à des domaines concrets, tels que : l'industrie, la robotique, la simulation informatique, la chimie/biologie moléculaire. Pour chacune de ces applications, voyons un peu plus en détail en quoi cela consiste.

Dans l'industrie textile, étant donné un motif (en maroquinerie, on dit aussi un patron) et une surface de matière première contenant des défauts, quelles sont les façons de découper un morceau de matière similaire au motif qui ne contienne aucun défaut ? Dans l'industrie métallurgique, un problème similaire survient. Ce problème est mentionné dans l'article de Chew et Kedem (1993). Il s'énonce comme suit : à partir d'une plaque de métal trouée par endroits, comment découper le plus grand carré de métal possible qui n'ait pas de trou ? Deux problèmes de la sorte peuvent être résolus automatiquement par le calcul si l'on sait traiter géométriquement les problèmes de placement correspondants.

En robotique, pour gérer les déplacements d'un robot on souhaite vérifier qu'une forme géométrique donnée, se déplaçant selon une certaine trajectoire, n'entre pas en collision avec des obstacles de position connue. Ce type de calcul est utile pour ce que l'on appelle la *planification de trajectoires cf.* les travaux de Leven et Sharir (1987b), Agarwal et al. (1997), Hirsch et Halperin (2002). La résolution de problèmes de placement, de manière à calculer non pas un mais tous les emplacements autorisés, permet là encore de répondre à cette question.

En simulation, avec des objets virtuels statiques, le placement permet de concevoir des structures de données pour accélérer les calculs testant les intersections entre objets géométriques. On parle de *détection de collisions cf.* le livre de Ericson (2005b) pour une présentation complète de ce thème. L'efficacité des algorithmes de détection de collisions est essentielle dans tout contexte temps-réel, dans les jeux vidéos ou la réalité virtuelle par exemple.

Finalement, la recherche en chimie appliquée a donné lieu à ce que l'on appelle dorénavant l'algorithmique moléculaire. L'émergence d'outils algorithmiques pour la biologie moléculaire remonte aux années 80, leur objectif est principalement de trouver, via une étude combinatoire poussée, des médicaments plus faciles à extraire ou à synthétiser. Un des problèmes bien étudiés en biologie moléculaire se nomme le *docking*. Ce dernier a pour objet l'étude de la complémentarité ligand-récepteur, qui tient compte de contraintes géométriques. Ce lien entre chimie et géométrie a donc donné lieu à des transferts de résultats entre la géométrie algorithmique et la biologie moléculaire *cf.* l'article de Laurie et Jackson (2006) et la thèse de Loriot (2008).

1.2.2 Travaux existants

De nombreux travaux résolvant des problèmes de placement existent dans la littérature. Dans la première partie du chapitre 5 nous en proposons une étude détaillée. On peut citer dès à présent les articles classiques de Chazelle (1983) et Canny (1989). *Grosso modo* tout problème de placement peut s'énoncer ainsi : étant donnée une forme géométrique et un environnement avec obstacles, comment placer la forme géométrique de sorte à satisfaire un objectif donné ? Par exemple, la forme géométrique peut être un polygone convexe et l'ensemble d'obstacles un nuage de points. L'objectif peut être simplement d'éviter les obstacles, ce qui revient à placer le polygone convexe donné de sorte qu'il n'y ait aucun point en son intérieur.

Chaque problème de placement décrit dans les travaux existants fait (plus ou moins explicitement) une hypothèse concernant la forme géométrique que l'on place. Concrètement, si l'on dispose d'un objet réel, on doit d'abord choisir comment représenter numériquement sa forme pour réaliser des calculs sur machine. Majoritairement, c'est un polygone (simple et/ou convexe) qui est utilisé. Durant cette thèse, nous avons exploré une autre voie en choisissant là encore une représentation par union de disques.

1.2.3 Placement d'union de disques dans un nuage de points

On trouve dans la littérature la description de plusieurs méthodes de placement, la plupart utilisant une forme à placer polygonale. Nous nous sommes quant à nous intéressés au contexte où il s'agit de placer une union de disques au sein d'un nuage de points. Dans la deuxième partie du chapitre 5, nous décrirons un *cadre global* (un formalisme et le vocabulaire associé) dans lequel on peut formuler de manière cohérente le placement d'union de disques quelque soient les transformations que l'on se donne pour réaliser ce placement.

Pour commencer, au début du chapitre 6, nous proposons une solution originale pour résoudre le problème relativement simple du placement d'union de disques par rotation uniquement.

Déjà, par rapport à la résolution de problèmes similaires que l'on trouve dans la littérature, notre solution offre plusieurs avantages notables :

- la solution proposée fonctionne sans limitation sur la complexité de la forme géométrique à placer. En particulier, celle-ci peut être non-convexe, non-symétrique, comporter un ou plusieurs trou(s) ;
- comme elle consiste en un traitement de données purement itératif et basé sur des objets géométriques, elle est d'une grande simplicité à mettre en œuvre. Elle ne nécessite aucun outil mathématique avancé, contrairement à certaines solutions mentionnées dans la littérature *cf.* livre de Sharir et Agarwal (2010) ;
- comme les disques constituant la forme à placer sont traités indépendamment les uns des autres, la parallélisation de l'algorithme est envisageable (nous en reparlerons dans la partie des perspectives) ;
- le résultat de l'algorithme consiste en un ensemble de régions simples ; une information compacte qui identifie l'ensemble tous les « emplacements libres » pour placer la forme géométrique. Les solutions sont alors connues exhaustivement.

1.2.4 Résolution générique de problèmes de placement

Dans la littérature, on constate que les méthodes de placement proposées permettent de résoudre un et un seul problème. Les solutions existantes partagent le fait d'être spécifiques à un problème.

Si le contexte change, par exemple : les transformations autorisées, la solution ne s'applique plus. Cela constitue une faiblesse dans la mesure où les problèmes de placement sont eux-mêmes nombreux et variés. Le modèle choisi est dépendant de l'application envisagée. *À notre connaissance,*

il n'existe aucune méthode de résolution unique qui puisse s'appliquer sur différentes variantes du placement de formes. Nous présentons une telle méthode dans ce mémoire.

Après avoir traité le placement d'union de disques par rotation et proposé un premier algorithme de résolution, nous avons généralisé ce dernier pour le rendre *générique*. La méthode obtenue est dite *générique* car elle peut opérer indifféremment sur plusieurs problèmes de placement. Le chapitre 6 expose cette méthode en détail.

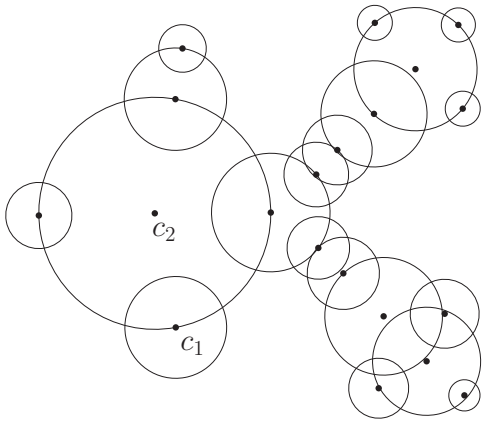
Avec cet outil, *nous avons effectivement résolu quatre problèmes* : placement par rotation, par homothétie, par translation, par homothétie et rotation combinées. Le chapitre 6 expose également les résultats respectifs obtenus.

Prenons par exemple le placement d'union de disques par rotation, que l'on illustre par la figure 1.8. Rappelons que pour une union de disques $\cup B$ donnée, une *copie* est obtenue par une combinaison de transformations parmi : translation, homothétie, rotation. Étant donné un nuage de points P, une union de disques $\cup B$, un point du plan u_k, un facteur d'échelle λ_k, le placement par rotation consiste à déterminer quelles sont les *copies* de $\cup B$, dont le point de référence et l'échelle sont égaux à u_k, λ_k respectivement, telles qu'elles ne contiennent aucun point de P en leur intérieur.

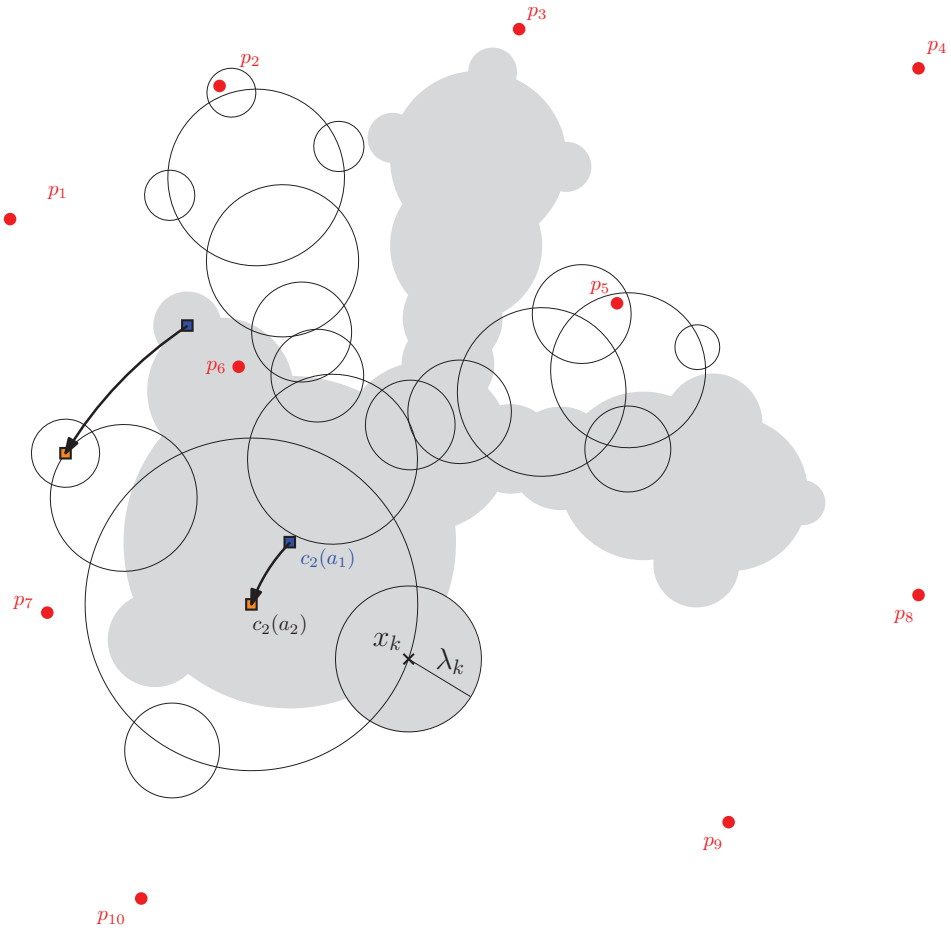
L'ensemble des solutions de ce problème prend la forme d'un ensemble d'arcs de cercle, comme l'illustre la figure 1.9.

Ces quatre problèmes sont résolus de la même manière pour l'exemple. Le procédé de calcul générique n'est pas limité à ceux-là mais s'applique à toute une classe de problèmes. Nous avons mis en évidence la condition suffisante pour pouvoir utiliser l'algorithme générique.

Remarque. L'ordre de lecture du présent mémoire n'est pas totalement contraint. Le chapitre 2 est un pré-requis pour tous nos travaux. Ensuite, les deux parties de la thèse sont indépendantes l'une de l'autre, bien que complémentaires. Au sein de chaque partie, les chapitres « intérieurs » se suivent. le chapitre 4 est l'approfondissement du principe et de la définition introduits au chapitre 3. Le chapitre 6 utilise le cadre global présenté au chapitre 5 et cherche à résoudre les problèmes qui y ont été formalisés. Il en constitue donc la suite logique.



(a) *ensemble de disque B utilisé pour l'exemple*



(b) *deux copies testées ne faisant pas partie des solutions du problème. On voit que le passage d'une copie à l'autre correspond à une simple rotation (représentée par les deux flèches en gras) autour du point de référence x_k (représenté par une croix)*

FIGURE 1.8 – *Problème du placement par rotation : étant donnés une union de disques et un nuage de points, ledit problème consiste à déterminer et à grouper toutes les copies telles que leur intersection avec le nuage de points soit vide*

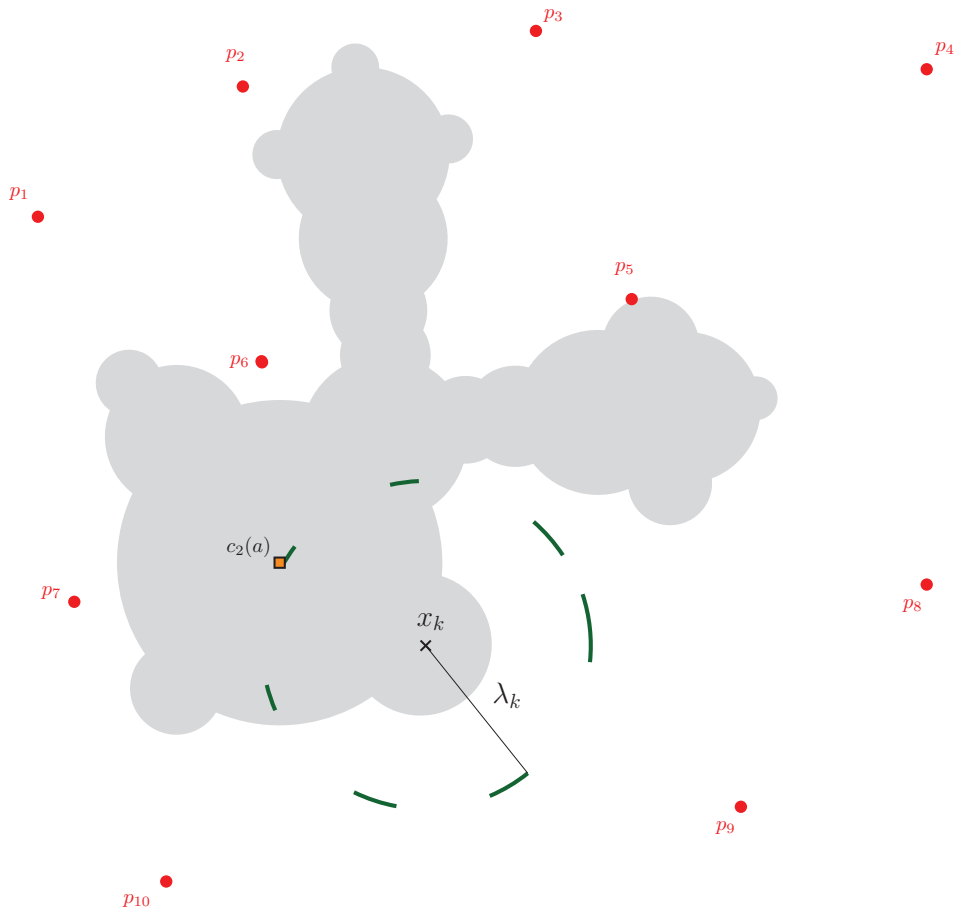


FIGURE 1.9 – *Exemple de solution au placement par rotation. Les arcs de cercles représentés en vert correspondent au lieu où si l'on place le centre du deuxième disque d'une copie (centre noté $c_2(a)$), alors la copie toute entière est vide de P*

2 Modèle à base de disques pour les formes géométriques planes

Sommaire

2.1 Représentation de formes géométriques à base de disques 21
2.2 Notations et définitions 24
2.2.1 Ensemble et union de disques standard 24
2.2.2 Notion de configuration 26
2.2.3 Copie paramétrée et fonction-centre 27
2.2.4 Copie vide 29
Conclusion du chapitre 29

Où l'on décrit un modèle commun à l'ensemble de nos travaux. Il s'agit de représenter les formes géométriques planes par une union de disques. À partir d'une telle union de disques, on définit des copies qui sont obtenues par combinaison de translation, homothétie, rotation. Ces copies sont déterminées par un point de référence, un réel positif qui représente sa taille (tout comme le disque), et une orientation. Nous présentons un formalisme et un vocabulaire associé. En ce sens, le contenu de ce chapitre est le « langage » commun à tous nos travaux.

2.1 Représentation de formes géométriques à base de disques

En géométrie algorithmique, une majorité de travaux utilisent un modèle polygonal comme représentation de formes géométriques complexes. La convexité (utiliser un polygone *convexe*) est parfois imposée dans le but d'obtenir des algorithmes plus efficaces. Dans la sous-section qui suit nous allons voir que certains chercheurs ont envisagé d'autres possibilités pour représenter les formes.

Dans certains domaines applicatifs tels que la détection de collision et la conception assistée par ordinateur (CAO), une des alternatives pour représenter les formes géométriques consiste à agencer plusieurs disques de rayons et positions différentes, pour définir une union de disques apte à représenter la forme géométrique souhaitée.

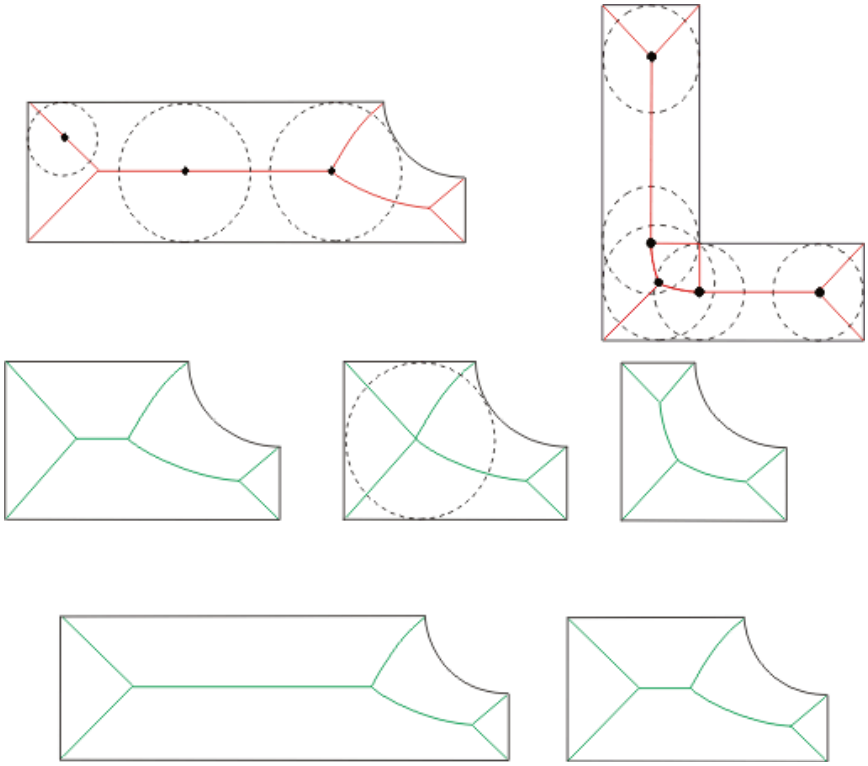
Les travaux de Hubbard (1996) recommandent l'usage d'une telle représentation à base de sphères ou de boules, pour résoudre de manière plus simple et efficace les problèmes de détection de collision. Ceux-ci surviennent dans les domaines tels que les jeux vidéos ou les simulations physiques macroscopiques. Cette recommandation est accompagnée de références aux algorithmes manipulant des unions de disques et de plusieurs arguments concernant les gains en termes de rapidité des calculs. L'auteur juge l'efficacité de cette représentation bonne voire supérieure au modèle polygonal/polyédral, pour le domaine de détection de collision.

Dans le même esprit, les travaux de thèse de Tam (2004) proposent une représentation appelée Voronoi Ball Models ou VBMs en tant que représentation multi-usages applicable aux domaines de l'informatique graphique, de la visualisation scientifique et de la vision par ordinateur. L'efficacité des VBMs est évaluée selon six critères : stabilité, flexibilité, précision, complexité, efficacité et caractère intuitif. L'auteur choisit ces critères qui ont, selon lui, un impact fort sur le champ d'application d'un modèle informatique pour les formes. L'auteur a également étudié la capacité des VBMs à gérer diverses opérations concernant la forme représentée :

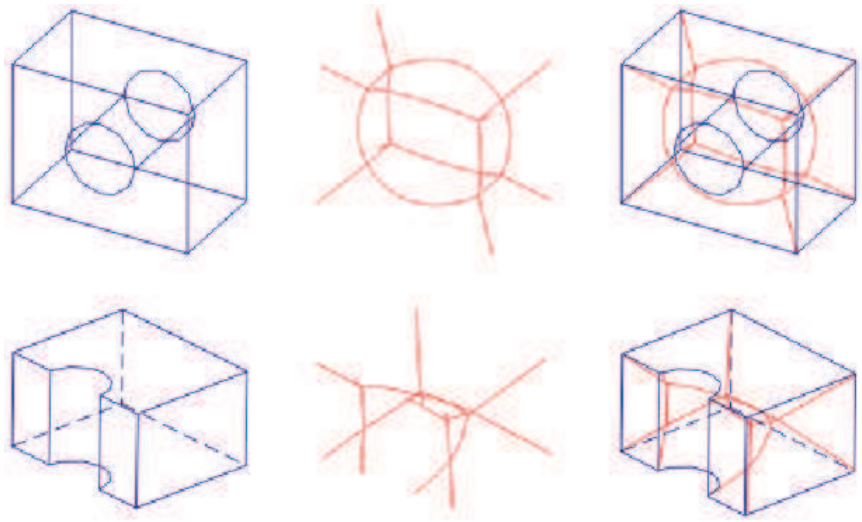
- extraction de forme (shape extraction)
- simplification de forme
- mise en correspondance (shape matching)
- interpolation
- manipulation et reconstruction de surface

Une telle représentation à base de disques peut être obtenue via une approximation discrète de la représentation nommée *représentation par axe médian* qui a été introduite dans le domaine de l'analyse d'images. On parle aussi de *squelette*. Des exemples d'une telle représentation sont donnés via la figure 2.1.

Remarquons aussi que les ensembles de disques ont été souvent utilisés par le passé dans le contexte de problèmes de *packing*. Le packing est une famille de problèmes liée à l'optimisation automatique, ceux-ci consistent à disposer un ensemble d'objets ayant une certaine forme géométrique en optimisant un certain critère (la densité le plus souvent). Pour l'exemple, un énoncé type de *packing problem* est le suivant : quel est le nombre maximal de cercles unités que l'on peut placer à l'intérieur d'un carré de côté $4+\sqrt{3}$? Un état de l'art de tels problèmes est donné par Ste-



(a) *cas 2d, représentation de sept formes différentes*



(b) *cas 3d, représentation de trois formes différentes avec de gauche à droite : l'objet, l'axe médian, la superposition des deux*

FIGURE 2.1 – *Exemples de représentation par axe médian d'une forme géométrique définie par son contour. Auteurs des illustrations : QUB -Finite Element Research Group 2012, http ://sog1.me.qub.ac.uk/*

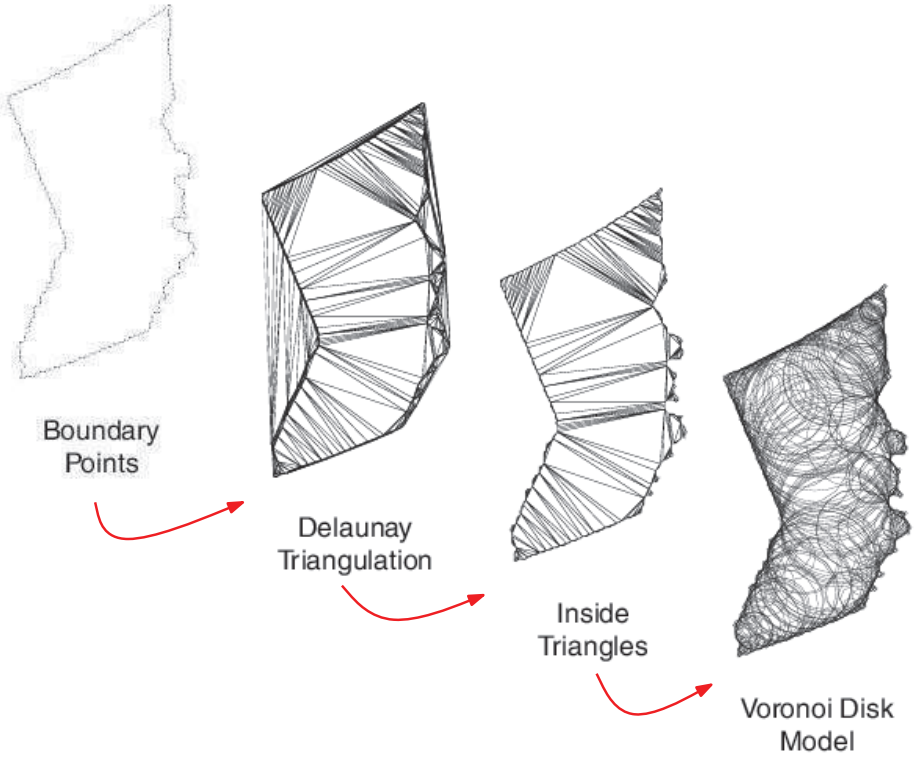


FIGURE 2.2 – *Méthode possible pour passer d'une représentation polygonale à union de disques, auteur de l'image : Tam (2004)*

phenson (2003). Les travaux présentés dans cette thèse ne sont pas directement liés au *packing*, aussi nous n'entrerons pas davantage dans les détails.

On peut mentionner au moins trois avantages d'ordre algorithmique, que nous avons repérés durant la conception et la production des programmes, quant à l'utilisation d'une représentation à base de disques.

- Une union de disques se décompose en éléments simples ce qui a pour effet de fournir des traitements itératifs et une conception algorithmique facilitée.
- chaque élément pouvant être traité indépendamment de la forme complète, le calcul parallélisé est possible.
- chaque élément peut être retiré ou modifié ce qui n'influence qu'une partie des résultats. De fait, la mise à jour de résultats pré-calculés est possible ce qui permet de faire des économies de temps de calcul.
- une union de disques est polyvalente en ceci qu'elle peut se substituer à des formes diverses et complexes, à condition que l'on autorise une légère approximation des contours. Les formes auxquelles on peut substituer une union de disques peuvent être concaves, contenir un ou plusieurs trous, *etc.*

Pour finir, notons que la génération d'une représentation basée sur les disques à partir d'autres représentations (en utilisant le squelette topologique notamment) est une question qui a déjà fait l'objet de plusieurs études, par exemple les articles de Tam et Fournier (2001), Amenta et al. (2001). Dans la figure 2.2 nous reprenons l'illustration d'un des articles pour montrer les étapes du passage d'un polygone à une union de disques.

Les sections qui suivent seront consacrées à présenter les définitions et notations, associées au modèle à base de disques que nous utiliserons dans les chapitres ultérieurs.

2.2 Notations et définitions

Soient $x_1, x_2 \in \mathbb{R}^2$ deux points du plan euclidien. Dans la suite, on utilise les notations suivantes. On note $\|x_1 - x_2\|$ la distance euclidienne entre x_1 et x_2 ; $[x_1x_2]$ le segment de droite ayant pour extrémités x_1 et x_2 ; $\overrightarrow{x_1x_2}$ le vecteur allant de x_1 à x_2 ; $\mathcal{O}$ l'origine du repère c'est-à-dire le point de coordonnées $(0,0)$; $\angle\left(\overrightarrow{\mathcal{O}x_1}, \overrightarrow{\mathcal{O}x_2}\right)$ l'angle fait par les vecteurs $\overrightarrow{\mathcal{O}x_1}$ et $\overrightarrow{\mathcal{O}x_2}$.

Dans nos travaux, nous avons utilisé des ensembles de disques pour modéliser toute forme géométrique plane. Un tel ensemble $B = \{b_1, \ldots, b_m\}$ est composé d'un nombre fini m de disques ouverts de $\mathbb{R}^2$. Chaque disque noté b_i ; $1 \leq i \leq m$ de cet ensemble est un disque ouvert de centre c_i et de rayon r_i. Formellement :

$$b_i = b(c_i, r_i) = \{x \in \mathbb{R}^2 \mid \|x - c_i\| < r_i\}$$

De plus, on emploie la notation ∂b_i pour désigner le *cercle* ayant des paramètres identiques à ceux de b_i, c'est-à-dire un centre c_i et un rayon r_i.

$$\partial b_i = \partial b(c_i, r_i) = \{x \in \mathbb{R}^2 \mid \|x - c_i\| = r_i\}$$

Dans les sous-sections suivantes, nous listons toutes les définitions employées par notre modèle.

2.2.1 Ensemble et union de disques standard

Nous représentons toute forme géométrique plane par un ensemble de disques. Pour pouvoir travailler sur cet ensemble de disques, nous avons convenu de trois conditions pré-requises. Pour distinguer un ensemble de disques quelconque d'un ensemble de disques qui satisfait ces conditions, nous définissons ci-dessous un nouveau terme.

Définition 2.2.1 (Ensemble de Disques Standard). On appelle Ensemble de Disques Standard (abrégé EDS) tout ensemble $B = \{b_i; i = 1, \ldots, m\}$ de disques ouverts de $\mathbb{R}^2$ qui vérifie les propriétés suivantes :

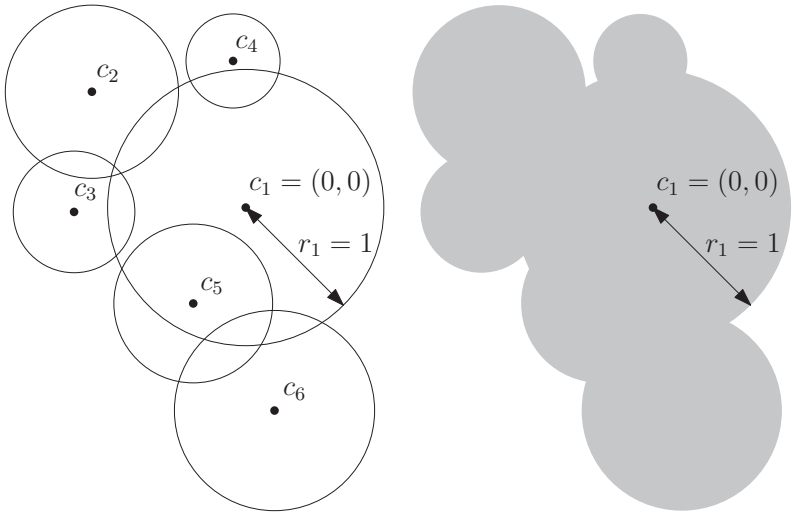
1. B contient le disque unité ;

$$b_1 = b(c_1, r_1) \text{ où } c_1 = \mathcal{O} = (0,0) \text{ et } r_1 = 1$$

2. B est le plus petit ensemble (*i.e.* ensemble de cardinalité minimale) dont l'union décrit la région du plan associée :

$$\forall B' \subsetneq B \text{ on a } \cup B' \neq \cup B$$

3. L'union $\cup B = \cup_{i=1}^{m} b_i$ constitue une région connexe.



(a) *exemple d'EDS, noté B à six disques* (b) *union de disques associée*

FIGURE 2.3 – *Ensemble de Disques Standard et union de disques*

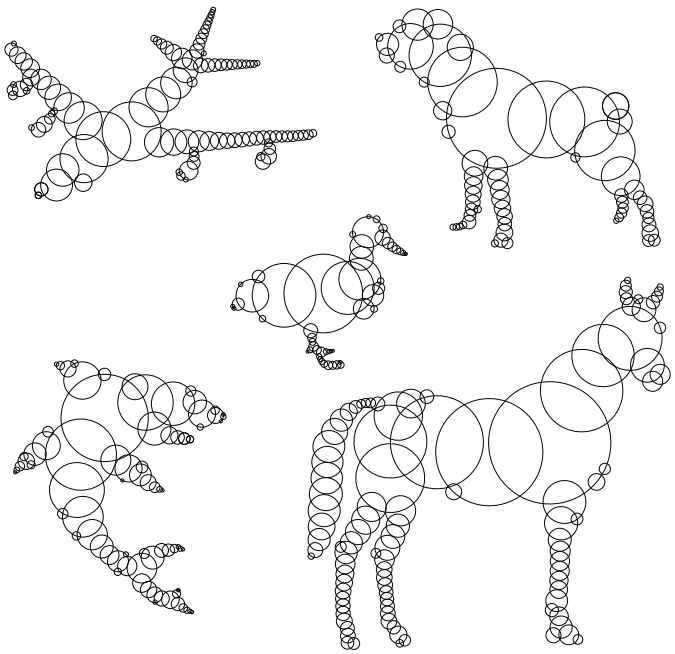


FIGURE 2.4 – *Représentation de formes géométriques complexes par union de disques*

A titre d'exemple, une union de disques est représentée par la figure 2.3a. Avec ce modèle, en variant le nombre de disques utilisés, on peut représenter aussi bien des formes géométriques simples que des formes beaucoup plus complexes. La figure 2.4 illustre l'« expressivité » du modèle choisi, c'est-à-dire sa capacité à représenter une grande diversité de formes géométriques.

Remarque. Les trois conditions de la précédente définition sont relativement peu contraignantes. Moyennant un traitement pouvant être automatisé, les deux premières peuvent être obtenues à partir d'un ensemble de disques quelconque. Le-dit traitement étant évident (il n'implique que des calculs basiques) nous n'avons pas jugé utile de le présenter ici.

Ces trois propriétés-conditions réduisent globalement la complexité des représentations. Cela facilite la conception d'algorithmes travaillant sur de telles formes et permet de rendre ceux-ci plus efficaces. En particulier, chacune des propriétés-conditions est justifiée pour la raison suivante. Condition 1 : on souhaite que la position et l'échelle de l'union de disques soient *normalisées*, car cela simplifie les calculs et définitions ultérieures. Condition 2 : on s'assure que le nombre de disques pour représenter une forme géométrique soit minimal, afin de ne pas ralentir inutilement les calculs. Ainsi, la durée de calcul pour les algorithmes utilisant l'union de disques peut être optimale, car il n'y a pas redondance de l'information. Condition 3 : la *connexité* de l'union de disques offre des possibilités supplémentaires pour accélérer les algorithmes résolvant des problèmes géométriques divers et utilisant l'union de disques. Plus précisément, la connexité permet une réduction *a priori* de la taille des jeux de données à considérer.

Un Ensemble de Disques Standard B a la capacité de modéliser diverses formes géométriques. La définition qui suit, directement liée à l'ensemble de disques standard, permet de faire la distinction entre les données utilisées (l'information des centres et rayons de disques) et la forme à modéliser.

Définition 2.2.2 (union de disques standard). Étant donné un EDS noté B, on appelle *union de disques* et on note $\cup B$, la région du plan euclidien obtenue par union ensembliste de tous les éléments de B. De plus, on note $\partial(\cup B)$ le bord de cette union.

Dans la suite de la thèse, par commodité, nous utiliserons le terme *union de disques* en lieu et place d'union de disques standard.

Pour utiliser cette représentation de formes en pratique et pouvoir résoudre des problèmes géométriques, nous avons conçu un formalisme dédié et avons défini des transformations géométriques pouvant être appliquées sur les unions de disques. Ainsi dans la sous-section suivante, nous introduisons les notions de configuration et de copie paramétrée, avec leurs notations respectives.

2.2.2 Notion de configuration

Plus loin dans ce chapitre nous allons introduire la notion de « copie ». Brièvement : une copie désigne une région obtenue par application d'une

certaine combinaison de transformations géométriques (parmi translation, homothétie, rotation) à une union de disques $\cup B$.

Dans la présente sous-section, nous introduisons d'abord la notion permettant d'identifier une combinaison de transformations.

Définition 2.2.3 (configuration). On appelle *configuration* tout triplet $a = (x, \lambda, \theta)$ élément de l'ensemble Ω défini par :

$$\Omega := (x, \lambda, \theta) \in \mathbb{R}^2 \times \mathbb{R}^+ \times [0, 2\pi[$$

Toute combinaison de translation, homothétie, rotation se caractérise donc par une *configuration* ; un triplet de paramètres où chacun correspond respectivement à : un vecteur $x \in \mathbb{R}^2$, un facteur d'agrandissement $\lambda \in \mathbb{R}^+$, et un angle de rotation $\theta \in [0, 2\pi[$.

Dans la sous-section qui suit, nous définissons plus en détail la notion de copie et explicitons son lien avec la notion de configuration.

2.2.3 Copie paramétrée et fonction-centre

On l'a dit, une copie désigne une région obtenue par application d'une combinaison de translation, homothétie, rotation, à une union de disques $\cup B$. Une telle combinaison est une transformation géométrique préservant la similarité.

On a vu que toute copie peut être caractérisée par une configuration et que toute configuration identifie une copie, c'est la raison pour laquelle on va utiliser une configuration a comme paramètre sur une union de disques $\cup B$. Il en résulte une définition concrète de la copie que l'on donne ci-dessous.

Définition 2.2.4 (copie paramétrée). Soient B un EDS et $a = (x, \lambda, \theta)$ une configuration. On appelle copie de B, notée $\cup B(a)$ la région produite par l'union des disques de l'ensemble $B(a)$, ensemble défini par :

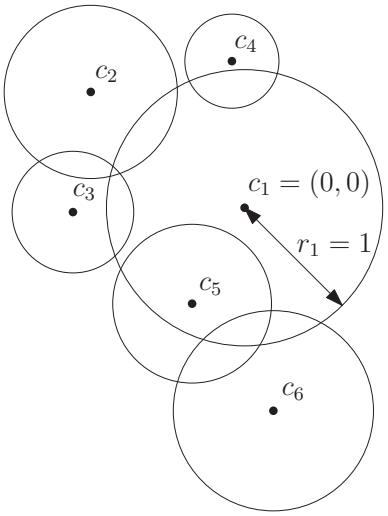
$$\begin{gathered} B(a) = \{b_1(a), \ldots, b_m(a)\} \\ \text{où } b_i(a) = b(c_i(a), r_i(a)) \text{ pour } 1 \leq i \leq m, \\ \text{avec } c_i(a) = c_i(x, \lambda, \theta) = x + \lambda \mathbf{M}_\theta c_i \text{ et } r_i(a) = \lambda r_i \\ \text{où } \mathbf{M}_\theta \text{ dénote la matrice de rotation d'angle } \theta \end{gathered}$$

De plus, le point $c_1(a) = x$ est appelé *point de référence* de la copie et le réel $r_1(a) = \lambda$ est appelé *taille de la copie*.

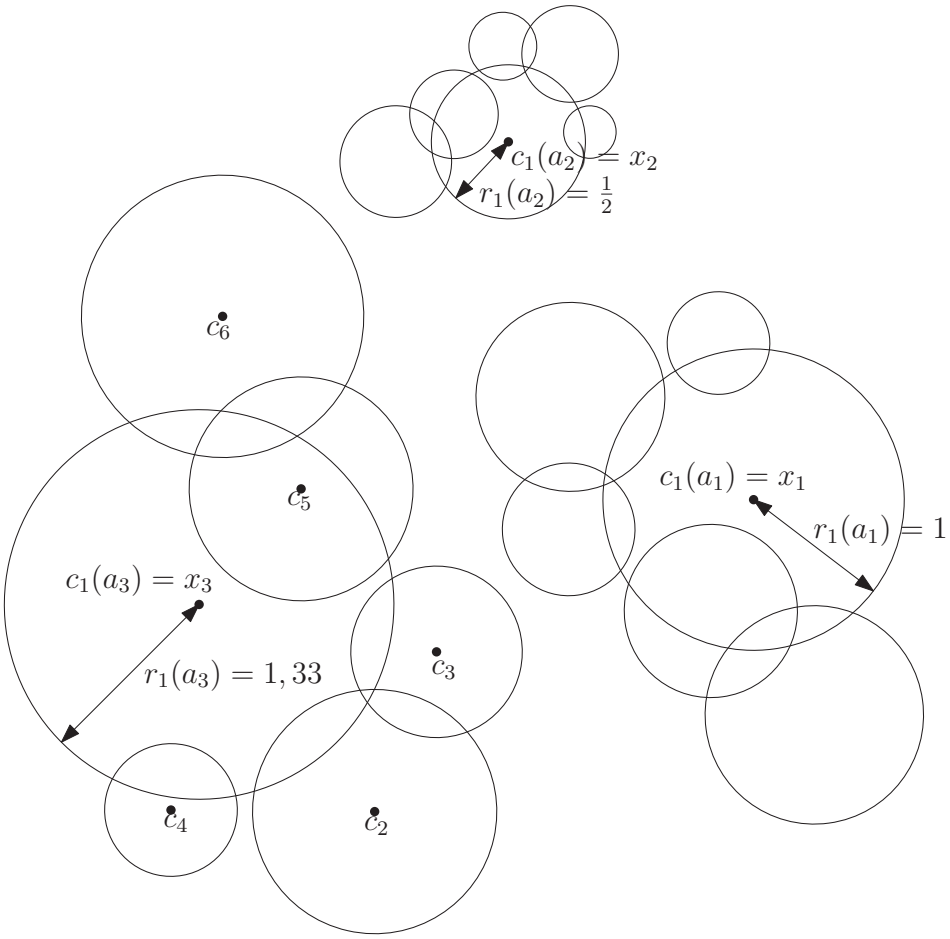
Un exemple de copie est illustré figure 2.5. Dans la définition qui vient d'être donnée, les coordonnées de $c_i(a)$ et les valeurs $r_i(a)$ sont définies afin de réaliser une transformation géométrique de toute l'union de disques $\cup B$. Comme dit précédemment, cette transformation consiste en l'application d'une ou plusieurs transformation(s) combinée(s) parmi : translation, homothétie, et rotation.

Remarques :

- Concernant la notation adoptée : pour faire une copie, il faut la donnée d'une configuration a et d'un EDS B indiquant la disposition originelle des disques (et modélisant ainsi une forme géométrique souhaitée). La configuration a indique la transformation géométrique à



(a) *Un Ensemble de Disques Standard (ou EDS en abrégé)*



(b) *Trois copies obtenues à partir du même EDS*

FIGURE 2.5 – *Notion de copies d'une union de disques. Pour chaque copie représentée, la position du premier centre et la valeur du premier rayon sont indiquées*

appliquer pour passer de l'union de disques $\cup B$ à la copie. Ceci explique notre choix de noter $\cup B(a)$ toute copie obtenue à partir de $\cup B$.
- Le symbole $\cup B(a)$ dénote une copie de $\cup B$ ayant pour configuration a. Le choix de ce formalisme est cohérent puisque la région $\cup B(a)$ est bien l'union de tous les disques de l'ensemble $B(a)$. On pourrait tout aussi bien écrire : $\cup_{i=1}^{m} b_i(a)$ en lieu et place de $\cup B(a)$.
- On insiste sur la différence entre Ensemble de Disques Standard (EDS) et copie paramétrée. Un EDS noté B modélise une certaine forme géométrique. Il s'agit d'une représentation canonique, d'une description abstraite ou archétype. *A contrario*, une copie paramétrée est une entité dont la localisation dans le plan euclidien est bien définie, elle possède une certaine taille et une certaine orientation.

Dans la définition de copie paramétrée qui vient d'être donnée, le symbole c_i dénote à la fois une constante et une fonction. La fonction c_i est appelée fonction-centre et permet de paramétrer la constante c_i par une configuration a.

Définition 2.2.5 (fonction-centre). On note c_i la fonction qui à une configuration $a = (x, \lambda, \theta)$ associe le point $c_i(a) = x + \lambda \mathbf{M}_\theta c_i$. Cette fonction est appelée fonction-centre.

Par abus de notation, nous utilisons le même symbole c_i pour désigner à la fois la fonction c_i prenant un paramètre a, et la constante c_i qui est le centre du i-ème disque de l'ensemble B. Le contexte dans lequel c_i est utilisé suffit à lever l'ambiguïté.

2.2.4 Copie vide

Définition 2.2.6 (copie vide). Étant considéré un ensemble fini $P = \{p_1, p_2, \ldots, p_n\}$ de points dans le plan euclidien, on appelle *copie vide par rapport à P* toute copie $\cup B(a)$ dont l'intersection ensembliste avec P est égale à l'ensemble vide. Formellement :

$$\cup B(a) \text{ est dite copie vide} \Leftrightarrow \cup B(a) \cap P = \emptyset$$

On remarque que la précédente condition équivaut aussi à l'assertion suivante. Pour tout couple : point $p \in P$, disque $b_i(a) \in B(a)$, le point p ne se trouve pas à l'intérieur du cercle $\partial b_i(a)$.

Conclusion du chapitre

Dans ce chapitre nous avons introduit plusieurs notions qui forment un socle commun à nos travaux. Les définitions : ensemble de disques standard, union de disques, copie, configuration, copie paramétrée et copie vide ont été fournies. Au chapitre suivant, on utilise une union de disques et ses copies pour proposer une nouvelle généralisation du diagramme de Voronoï.

Première partie

Diagramme de Voronoï et généralisation basée sur une union de disques

Nouvelle généralisation du diagramme de Voronoï

3

Sommaire

3.1 Rappels sur les régions de Voronoï 35
3.2 Généralisations du diagramme de Voronoï 35
3.2.1 Motivations 35
3.2.2 Généralisations existantes 36
3.3 Principe de la nouvelle généralisation 39
3.3.1 Prédicat du disque vide 39
3.3.2 Extension du prédicat aux unions de disques 41
3.3.3 Régions de Voronoï Paramétrées par une Union de Disques 41
Conclusion du chapitre 48

Où nous introduisons le concept de région de Voronoï Paramétrée par une Union de Disques, qui généralise le concept de région de Voronoï.

Nous commençons par quelques rappels sur le diagramme de Voronoï lui-même, ses diverses applications, les généralisations existantes décrites dans la littérature.

Ensuite, nous expliquons en détail le principe de généralisation que nous proposons. On part du constat que toute région de Voronoï classique vérifie le *prédicat du disque vide*. Nous étendons ce prédicat à une union de plusieurs disques. Une telle union est apte à représenter des formes géométriques variées, éventuellement complexes.

En nous basant sur le *prédicat d'union de disques vide* ainsi obtenu, nous définissons les nouvelles régions du plan appelées : régions de Voronoï Paramétrées par une Union de Disques (ou régions de VPUD). Nous explicitons comment celles-ci sont liées aux régions de Voronoï classiques. De nombreux exemples et schémas sont fournis pour aider à la compréhension.

3.1 Rappels sur les régions de Voronoï

Formalisé dès 1908 par le mathématicien russo-ukrainien Georgy Voronoï, le diagramme qui porte son nom a fait depuis l'objet de très nombreuses études. Pour un état de l'art détaillé, le lecteur non familier avec ces travaux peut se référer à Aurenhammer et Klein (2000). Dans le cadre de cette thèse, nous travaillons dans le plan euclidien et le diagramme de Voronoï se compose donc d'un ensemble de régions planes. Cette section a pour objet de rappeler la définition de ces régions et de quelques-unes de ses conséquences immédiates. Nous donnons aussi des exemples visant à montrer le vaste potentiel applicatif de la notion de région de Voronoï.

Dans le plan euclidien, on appelle région de Voronoï la région plane définie comme suit.

Définition 3.1.1 (région de Voronoï). Soit $P = \{p_1, \ldots, p_n\}$ un ensemble de points du plan euclidien appelés *germes*. À chaque germe p_i est associé une région de Voronoï définie comme l'ensemble des points plus proches ou aussi proches de p_i que de tout autre germe. Formellement :

$$V(p_i) := \left\{ x \in \mathbb{R}^2 \mid \|x - p_i\| \leq \left\|x - p_j\right\|, \forall p_j \in P \right\}$$

3.2 Généralisations du diagramme de Voronoï

Cette section a pour but de rendre compte des nombreuses généralisation déjà existantes du diagramme de Voronoï. Nous commençons par nous interroger sur la raison d'être de ces travaux, au-delà de la curiosité mathématique. Quelles ont été les raisons qui ont mené à une telle diversité de généralisation du diagramme de Voronoï ?

Nous proposons une réponse à cette question dans la première sous-section, avant de donner, dans la seconde sous-section, une liste d'exemples de généralisation et que l'on compare les unes aux autres en proposant trois grandes classes de diagrammes de Voronoï généralisés.

3.2.1 Motivations

Pourquoi généraliser le diagramme de Voronoï ? Comme cela a déjà été dit en introduction à la thèse, ce diagramme possède de nombreuses applications. Rappelons-en quelques-unes. Le livre de Okabe et al. (2000) résume les applications existantes dans de nombreux domaines concrets, nous reprenons ici quelques exemples (liste non exhaustive) :

- en astronomie, le diagramme de Voronoï est employé pour identifier des amas d'étoiles et de galaxies
- en sciences de l'environnement, les régions de Voronoï servent à modéliser la croissance et la compétition d'espèces végétales d'un écosystème (compétition pour un occuper un espace vital permettant la croissance végétale)
- en reconnaissance de formes (*pattern recognition* en anglais), un descripteur de formes unidimensionnel portant le nom de « squelette » est extrait à partir de formes géométriques 2D, grâce à la recherche de contours et à un traitement basé sur le diagramme de Voronoï

- en informatique, pour reconstruire une forme $3d$ à partir d'un nuage de points qui échantillonne la surface. Ce procédé est utile par exemple lors de la numérisation d'un objet physique à l'aide d'un capteur laser. Chaque mesure donne lieu à un point tridimensionnel, on cherche ensuite à reproduire fidèlement la surface sur ordinateur.
- en robotique, le diagramme de Voronoï couplé à des notions de théorie des graphes fournit une solution simple et élégante à la planification de trajectoires (*motion planning*). La planification de trajectoires permet à un robot mobile de se déplacer tout en évitant des obstacles physiques présents dans l'environnement

Dans le domaine de la géométrie algorithmique elle-même, plusieurs applications existent, dont la plus classique est : la résolution du *Largest Empty Problem cf.* article de Toussaint (1983).

Dans ce contexte, étendre la définition du diagramme de Voronoï permettrait d'augmenter encore ses applications. Cela permettrait d'accroître encore la portée de cet outil. Cette vision est défendue dans l'état de l'art de Aurenhammer et Klein (2000) qui liste les généralisations du diagramme de Voronoï et les applications associées. Ainsi, il existe par exemple une généralisation qui permet de résoudre le « Largest Empty Square problem », problème pour lequel le diagramme de Voronoï classique est insuffisant. Des applications plus nombreuses constituent ainsi une des motivations pour généraliser le diagramme de Voronoï.

3.2.2 Généralisations existantes

Dans cette partie, nous donnons un aperçu des très nombreuses généralisations du diagramme de Voronoï que l'on trouve dans la littérature Lee (1980), Lee et Drysdale (1981), Klein (1988). À l'origine le diagramme de Voronoï est défini :

- dans le plan euclidien,
- pour un ensemble de points,
- en utilisant la distance euclidienne.

La majorité des généralisations existantes consistent en l'extension de l'une de ces trois caractéristiques. En comparant les travaux existants entre eux, on conçoit alors trois catégories pouvant classer la plupart des généralisations de Voronoï. Nous décrivons ces catégories dans la liste ci-dessous, en donnant quelques exemples pour chacune d'elles.

- **Généralisation utilisant la notion de cellule.** Les cellules sont définies de manière très similaire aux régions de Voronoï, mais dans des espaces différents qui peuvent être de dimension supérieure à deux, voir par exemple l'article de Voronoï (1908) lui-même qui a étudié le cas général d'un « diagramme » en dimension n. La figure 3.1 illustre cette catégorie via l'exemple de régions de Voronoï dans l'espace tridimensionnel. Seules les arêtes des cellules sont représentées.
- **Généralisation utilisant des germes non ponctuels.** Il existe des variantes du diagramme de Voronoï, qui pour les germes, utilisent des objets géométriques qui diffèrent de simples points. On trouve la description de nombreuses telles variantes dans l'état de l'art de Aurenhammer (1991). Les germes peuvent être : des points pondérés (*weighted Voronoï diagram*) : articles de Karavelas et Yvinec (2002),

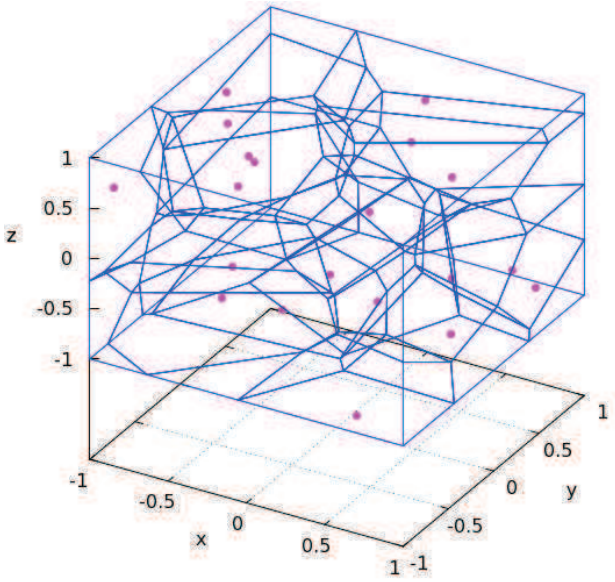


FIGURE 3.1 – *Généralisation du diagramme de Voronoï à l'espace* 3D

Aurenhammer (1987), des segments de droite (*line segments Voronoï diagram*) : article de Burnikel et al. (1994), des polygones convexes : article de Karavelas et Yvinec (2003), *etc.* La figure 3.2 illustre cette catégorie de généralisations via l'exemple d'un diagramme de Voronoï généralisé pour lequel les germes sont des segments de droite au lieu de simples points.

– **Généralisation utilisant une distance non-euclidienne.** La distance à considérer entre les germes peut être remplacée par une autre métrique. La nouvelle métrique utilisée peut être la norme L_1, L_∞ (elles font partie d'une famille de fonctions distance que l'on appelle les normes de Minkowski), ainsi que d'autres distances convexes comme expliqué dans les travaux de Chew et Drysdale III (1985). Récemment, les pseudo-métriques ou *divergences de Bregman* ont été utilisées par Nielsen et Nock (2011) pour définir une autre généralisation des régions de Voronoï. Les figures 3.3 et 3.4 illustrent cette catégorie via les exemples de, respectivement, régions utilisant la norme L_1 aussi appelée distance de Manhattan et régions utilisant une divergence de Bregman.

Ces trois catégories nous donnent un bon aperçu de la diversité des généralisations existantes, bien que quelques résultats dépassent cette simple classification. Par exemple, les diagrammes abstraits définis par Klein (1988) se concentrent davantage sur la frontière des régions que sur leur définition en tant que cellules.

Durant nos travaux, nous avons proposé une généralisation qui n'entre pas dans ces catégories. Cette généralisation est fondée sur l'extension d'une propriété connue, propre à toute région de Voronoï. Nous dévoilons cette nouvelle généralisation du diagramme de Voronoï à la section suivante.

FIGURE 3.2 – *Généralisation du diagramme de Voronoï aux segments*

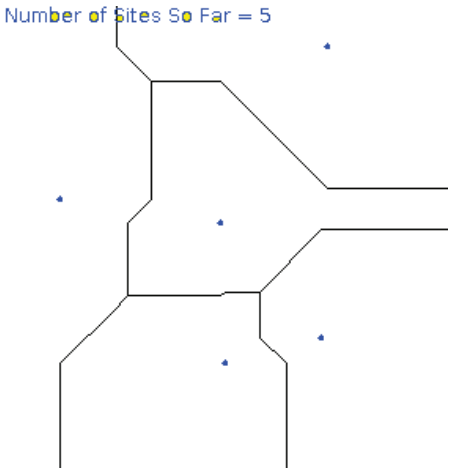


FIGURE 3.3 – *Généralisation du diagramme de Voronoï à la métrique L_1, aussi connue sous le nom de distance de Manhattan*

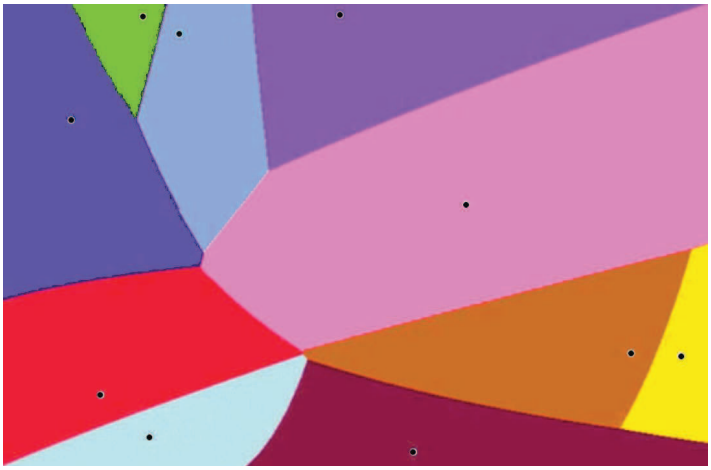


FIGURE 3.4 – *Généralisation du diagramme de Voronoï à une divergence de Bregman arbitrairement choisie.*

3.3 Principe de la nouvelle généralisation

La généralisation que nous allons décrire ici a fait l'objet de plusieurs communications scientifiques *cf.* Iwaszko et al. (2009a;b; 2010).

Elle offre plusieurs avantages : en étendant une propriété du diagramme de Voronoï à des formes géométriques plus complexes, elle permet de grouper, de structurer en régions un ensemble de formes vides. Ces formes étant représentées par une union de disques, une partie des calculs se ramène à ceux du diagramme de Voronoï classique. Ceci facilite la conception algorithmique en donnant la possibilité de s'inspirer ou de modifier les algorithmes déjà existants pour Voronoï classique.

Dans cette section, nous introduisons des régions particulières du plan euclidien, que nous appelons régions de Voronoï Paramétrées par une Union de Disques (ou régions de VPUD).

Notre idée de généralisation se résume en deux étapes. Premièrement, nous définissons le *prédicat du disque vide* qui peut s'appliquer à tout point du plan. Comme on va le voir, celui-ci fournit une manière alternative de définir toute région de Voronoï classique. Deuxièmement, nous étendons ce prédicat du disque vide à un cas plus général, en reprenant le modèle à base de disques déjà présenté au chapitre 2. Ces deux étapes ont été à l'origine de notre définition des régions de VPUD. Elles permettent de mieux comprendre le fait qu'une région de VPUD est une généralisation d'une région de Voronoï classique. Elles aident également à déduire les conséquences directes de la définition.

3.3.1 Prédicat du disque vide

La région de Voronoï $V(p_i)$ peut être décrite via un prédicat géométrique particulier. Un tel prédicat est intéressant car il permet d'aborder le diagramme de Voronoï sous un autre angle, en employant une *forme géométrique* simple : le disque plutôt que les fonctions distance. Nous définissons le prédicat comme suit.

Définition 3.3.1 (prédicat du disque vide). L'intersection entre P et le disque ouvert centré en x, de rayon égal à la distance entre x et p_i est vide. Ce prédicat s'exprime simplement via les notations ensemblistes usuelles :

$$P \cap b(x, \|x - p_i\|) = \emptyset$$

La région de Voronoï $V(p_i)$ peut être exprimée en utilisant le prédicat de disque vide :

$$V(p_i) = \left\{x \in \mathbb{R}^2 \mid P \cap b(x, \|x - p_i\|) = \emptyset\right\}$$

Cela constitue une *définition alternative* à la définition classique. L'appartenance à une région de Voronoï d'une part, la vérification du prédicat de disque vide d'autre part sont donc deux choses tout à fait équivalentes.

La figure 3.5 illustre cette définition alternative à l'aide de multiples exemples. Sur ces figures on voit un ensemble arbitraire de dix germes. La région de Voronoï générée par le germe p_1 est représentée en bleu clair. Considérant successivement six points du plan $x_k; 1 \leq k \leq 6$, l'appartenance de x_k à la région de Voronoï $V(p_i)$ coïncide à chaque fois avec

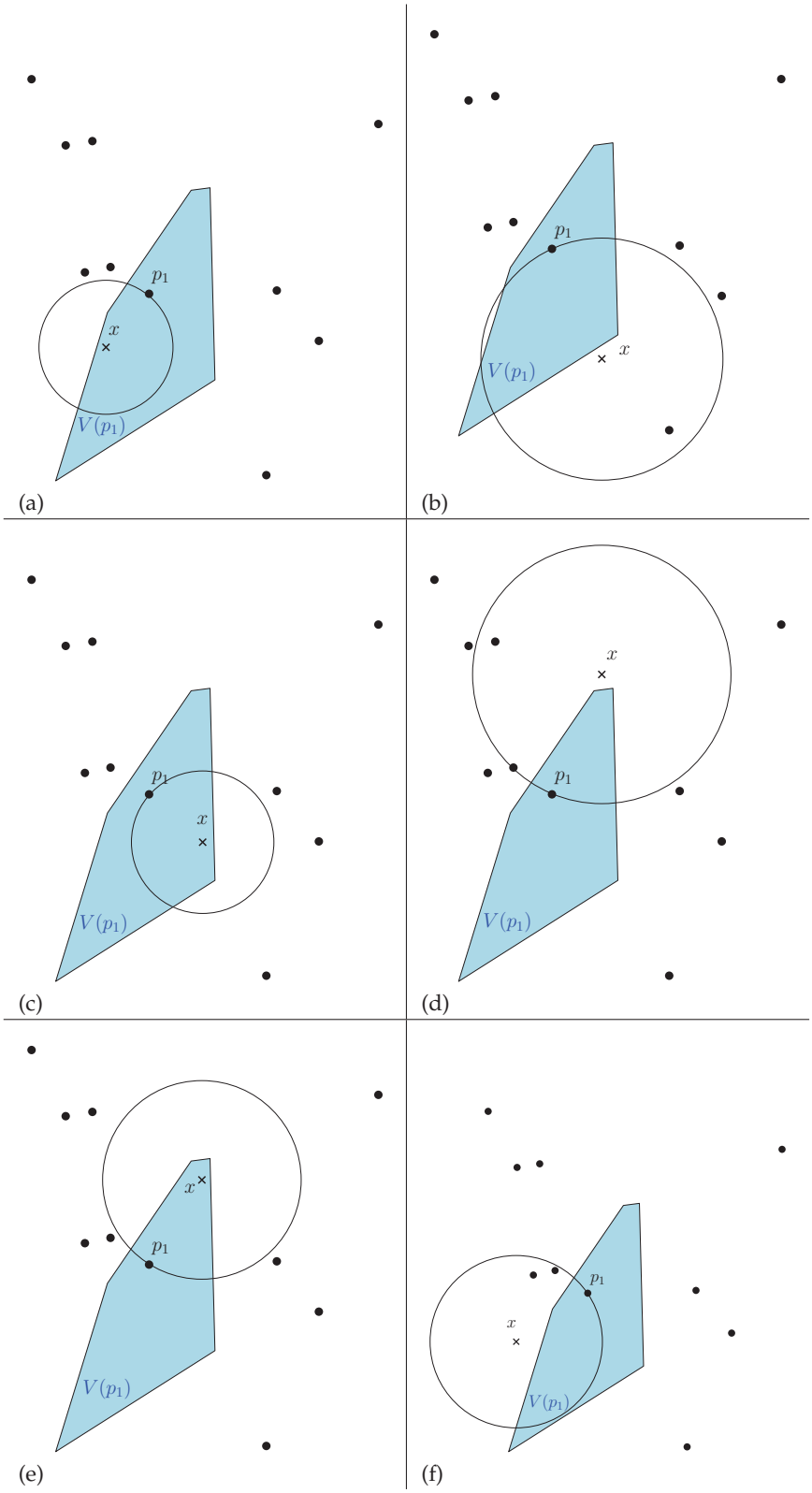


FIGURE 3.5 – *Équivalence entre appartenance d'un point x à une région de Voronoï $V(p_1)$, et vérification du prédicat de disques vide pour x, p_1. Le prédicat s'énonce : le cercle centré en x de rayon $\|x - p_1\|$ ne contient aucun site de P*

la vérité du prédicat pour P, p_i, x_k. Ces exemples sont représentés dans la colonne de gauche. Inversement, la fausseté du prédicat pour P, p_i, x_k coïncide avec la non-appartenance de x à la région $V(p_i)$. Ces exemples sont représentés dans la colonne de droite.

Cette mise en évidence du disque au sein de la définition des régions de Voronoï classiques est intéressante, dans la mesure où elle permet lier le diagramme de Voronoï à la notion de *forme géométrique* (le disque est un cas particulier de forme géométrique).

On peut alors s'interroger sur l'existence d'une variante du diagramme de Voronoï, impliquant des formes géométriques plus complexes. Par exemple : existe t-il un diagramme généralisé où les points de chaque région respectent un prédicat de *deux disques* vides ? Dans la sous-section suivante, nous verrons qu'une telle extension du prédicat de disque vide est possible, non seulement possible pour $m = 2$ disques mais pour tout $m \in \mathbf{N}, m > 1$.

3.3.2 Extension du prédicat aux unions de disques

La définition 3.3.1 du prédicat du disque vide a été donnée plus tôt. On a vu que ce prédicat suffit à définir ce qu'est une région de Voronoï.

Dans cette thèse, nous avons proposé d'étendre ce prédicat à un nombre quelconque, fini de disques. En reprenant les définitions du chapitre 2, le prédicat du disque vide peut effectivement être étendu à l'union des disques d'un Ensemble de Disques Standard (EDS). Le résultat de notre démarche est un *prédicat d'union de disque vide*, que l'on définit ci-dessous.

Définition 3.3.2 (prédicat d'union de disques vide). Soit une copie $\cup B(a)$ ayant pour configuration un triplet de la forme $a = (x, \|x - p_i\|, 0)$, c'est-à-dire : point de référence $x \in \mathbb{R}^2$, taille égale à la distance $\|x - p_i\|$ et orientation nulle $\theta = 0$. Le prédicat d'union de disques vide s'énonce ainsi : l'intersection entre $\cup B(a)$ et le nuage de points P est vide. Formellement :

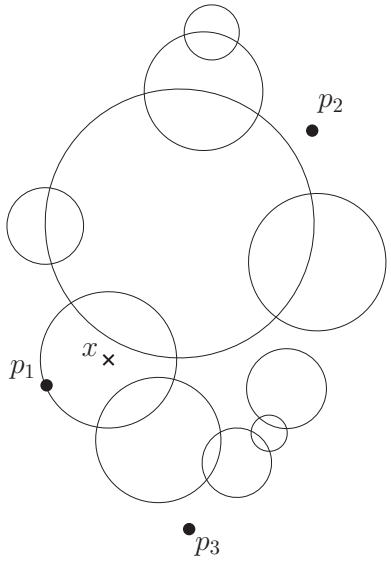
$$[\cup B(a)] \cap P = \emptyset \text{ avec } a = (x, \|x - p_i\|, 0)$$

Ce nouveau prédicat est général, nous l'illustrons sur la figure 3.6. Rappelons que, comme expliqué au chapitre 2, une union de disques peut représenter des formes géométriques complexes. De plus, un disque unique étant un cas particulier d'une union de disques, le *prédicat d'union de disques vide* généralise bien le *prédicat de disque vide*.

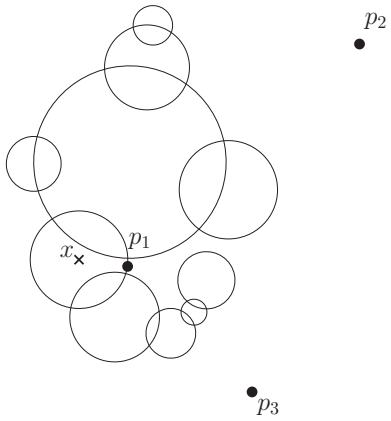
Procédant à partir de cette extension du prédicat de disque vide, nous avons défini des régions de Voronoï généralisées correspondantes. Leur définition est exposée dans la sous-section suivante.

3.3.3 Régions de Voronoï Paramétrées par une Union de Disques

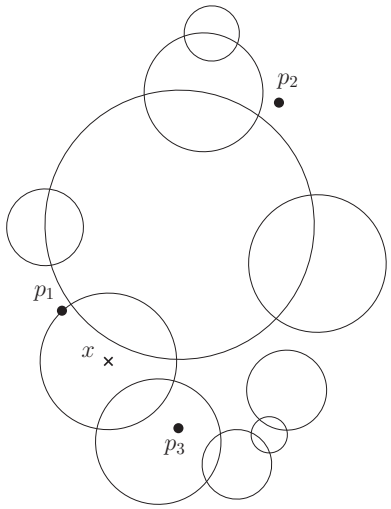
Au chapitre 2, nous avons donné la définition 2.2.1 d'ensemble de disques standard, ainsi que la définition 2.2.4 des copies. Nous avons ensuite expliqué qu'une forme géométrique quelconque est représentable par une telle union de disques. Dans les sous-sections 3.3.1 et 3.3.2, nous avons présenté le prédicat de disque vide, propre à toute région de Voronoï classique, et l'avons étendu à une union de disques. En nous basant



(a)



(b)



(c)

FIGURE 3.6 – *Prédicat d'union de disques vide : pour une forme géométrique représentée par union de plusieurs disques, un point x et un germe p_1, le prédicat considère une copie ayant son point de référence en x et de taille égale à la distance $\|p_1 - x\|$. Selon les arguments du prédicat, la-dite copie peut être vide (a,b) ou non-vide (c)*

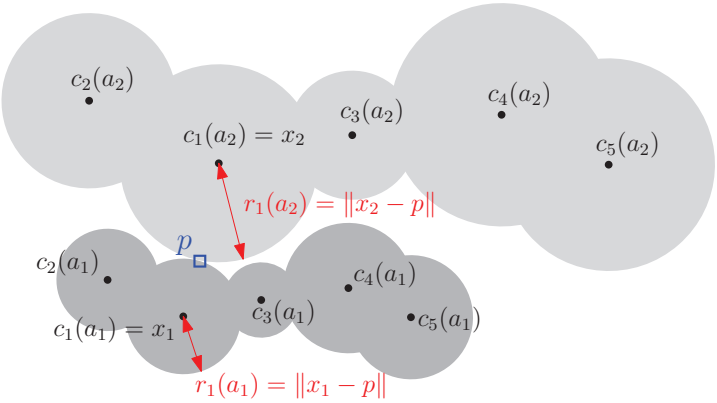


FIGURE 3.7 – *Exemples de copies considérées par le prédicat d'union de disques vide. Toutes sont liées par la contrainte suivante : la taille de copie doit être égale à la distance entre le point de référence x et un germe p (indiqué ici par un carré)*

sur tous ces travaux préliminaires, nous avons défini de nouvelles régions de Voronoï généralisées. Nous en donnons la définition ci-dessous.

Définition 3.3.3 (Région de Voronoï Paramétrée par une Union de Disques). Étant donnés un ensemble de disques standard $B = \{b_1, \ldots, b_m\}$ à m disques et un ensemble de points $P = \{p_1, \ldots, p_j, \ldots, p_n\}$, à tout point $p_i \in P$ on associe la région VPUD notée $V_B(p_i)$ et définie telle que :

$$V_B(p_i) := \{x \in \mathbb{R}^2 \mid \cup B(a) \cap P = \emptyset, \text{ avec } a = (x, \|x - p_i\|, 0)\}$$

Interprétation de la définition

Rappelons quelques définitions données au chapitre 2. On note $B = \{b_1, \ldots, b_m\}$ un Ensemble de Disques Standard (EDS), comme expliqué dans la définition 2.2.1. Étant donnée une configuration a comme expliqué dans la définition 2.2.3, l'ensemble $B(a) = \{b_i(a); 1 \leq i \leq m\}$ est un ensemble de disques obtenu en appliquant une même transformation géométrique sur tous les disques $b_i, 1 \leq i \leq m$. La transformation en question combine translation, homothétie et rotation. Elle est entièrement déterminée par la configuration $a = (x, \lambda, \theta)$. Plusieurs exemples de ce procédé de paramétrisation sont donnés dans la section 2.2.3.

Le symbole $\cup B(a)$ dénote une copie de $\cup B$ ayant pour configuration a. Le choix de ce formalisme est cohérent puisque la région $\cup B(a)$ est bien l'union de tous les disques de l'ensemble $B(a)$. On pourrait tout aussi bien écrire : $\cup_{i=1}^{m} b_i(a)$ en lieu et place de $\cup B(a)$.

Dans la définition 3.3.2 (prédicat d'union de disques vide), paramétrer $\cup B$ par $a = (x, \|x - p_i\|, 0)$ cela signifie que le point de référence (centre du premier disque) devient x, la taille de la copie augmente ou diminue jusqu'à avoir p_i sur la frontière du premier disque. Le troisième élément de la configuration, ici égal à 0, signifie qu'aucune rotation n'est appliquée à $\cup B$. La figure 3.7 donne deux exemples de copies dont les configurations respectives respectent cette contrainte : $a = (x, \|x - p_i\|, 0)$.

Ainsi, $V_B(p_i)$ représente l'ensemble des points $x \in \mathbb{R}^2$ où la copie $\cup B(x, \|x - p_i\|, 0)$ a une intersection vide avec P. Nous avons considéré plusieurs exemples pour illustrer cette nouvelle définition, les diagrammes

correspondants à trois jeux de données différents sont donnés sur les figures 3.8, 3.9 et 3.10.

Prédicat d'union de disque vide et rotation

Dans le cas du diagramme de Voronoï classique, on a vu que toute région satisfait le prédicat du disque vide. Or un unique disque est invariant par rotation autour de son centre. La question des rotations ne se pose pas.

Pour définir le prédicat d'union de disques vide, nous avons fait le choix d'ignorer la rotation *i.e.* de se donner une orientation fixe de l'union de disques. De fait, la généralisation du diagramme de Voronoï proposée consiste en un ensemble de régions pour lesquelles l'union de disques est vide, sans rotation aucune. Ce choix est motivé par le souhait de pouvoir représenter exactement la région généralisée en deux dimensions (comme pour le diagramme de Voronoï classique) ce qui ne serait pas possible si l'on tenait compte effectivement des rotations.

Si l'on souhaite appliquer le diagramme de VPUD à un domaine particulier où la rotation est indispensable, il est toujours possible d'échantillonner l'intervalle $[0, 2\pi[$ et de calculer le diagramme de VPUD pour différentes valeurs de θ dans cet intervalle.

Conséquences immédiates de la définition

- **Le diagramme de VPUD est une généralisation du diagramme de Voronoï.** Notons $B^{(1)} = \{b_1\}$ le singleton contenant le disque unité $b_1 = b(\mathcal{O}, 1)$. $\mathcal{O}$ est l'origine du repère. En considérant ce cas particulier où $B = B^{(1)}$, la définition 3.3.3 d'une région de VPUD devient :

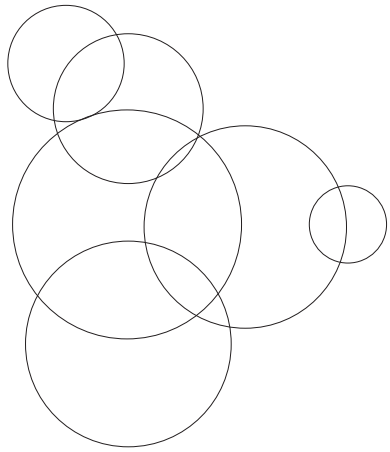
  $$V_{B^{(1)}}(p_i) = \left\{x \in \mathbb{R}^2 \mid b(x, \|x - p_i\|) \cap P = \emptyset\right\}$$

  Ce qui, comme on l'a vu plus tôt, définit une région de Voronoï classique. Le *prédicat d'union de disques vide*, généralise bien le *prédicat de disque vide*. Par suite des définitions, le diagramme de VPUD tout entier est bien une généralisation du diagramme de Voronoï.
- **Une région de VPUD ne peut être vide.** Nous nous sommes posés la question : existe t-il un cas dégénéré tel qu'une région de VPUD soit égale à l'ensemble vide ? La réponse est négative : quelque soit le triplet (P, p_i, B) que l'on choisit, triplet dont les coordonnées sont, respectivement un ensemble de points, un germe, un ensemble de disques standard, le germe p_i appartient à la région $V_B(p_i)$.
  Partant de la définition 3.3.3 d'une région de VPUD, on sait que :

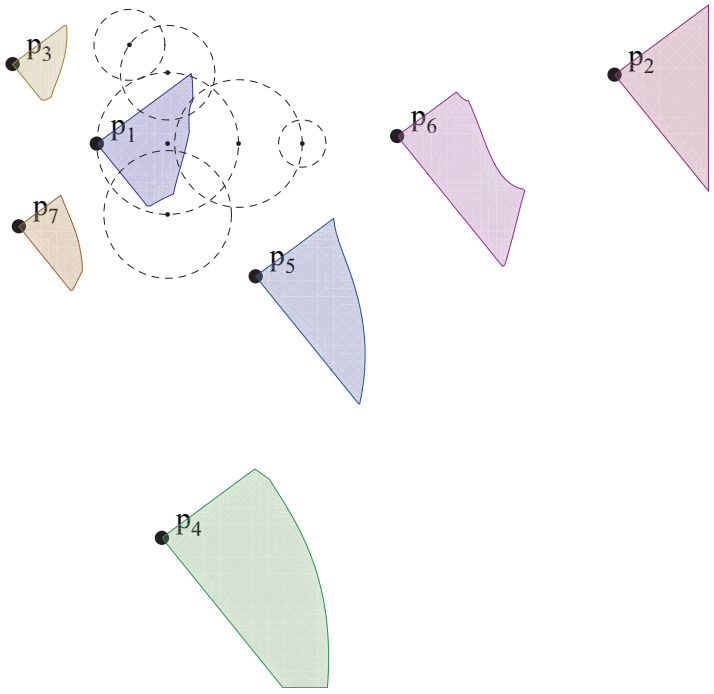
  $$x \in V_B(p_i) \Leftrightarrow \cup B(x, \|x - p_i\|, 0) \cap P = \emptyset$$

  Dans le cas où $x = p_i$, $\|x - p_i\| = \|p_i - p_i\| = 0$. Tout disque b_j de B étant un disque *ouvert* il vient que :

  $$\begin{aligned} b_j(x, 0, 0) &= b(x + 0\mathbf{M}_0 c_j, 0 r_j) \\ &= b(x, 0) \\ &= \{y \in \mathbb{R}^2 \mid \|x - y\| < 0\} \\ &= \emptyset \end{aligned}$$

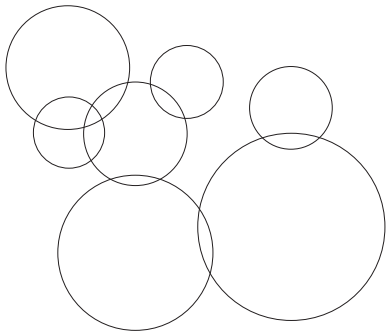


(a) *ensemble de disques considéré*

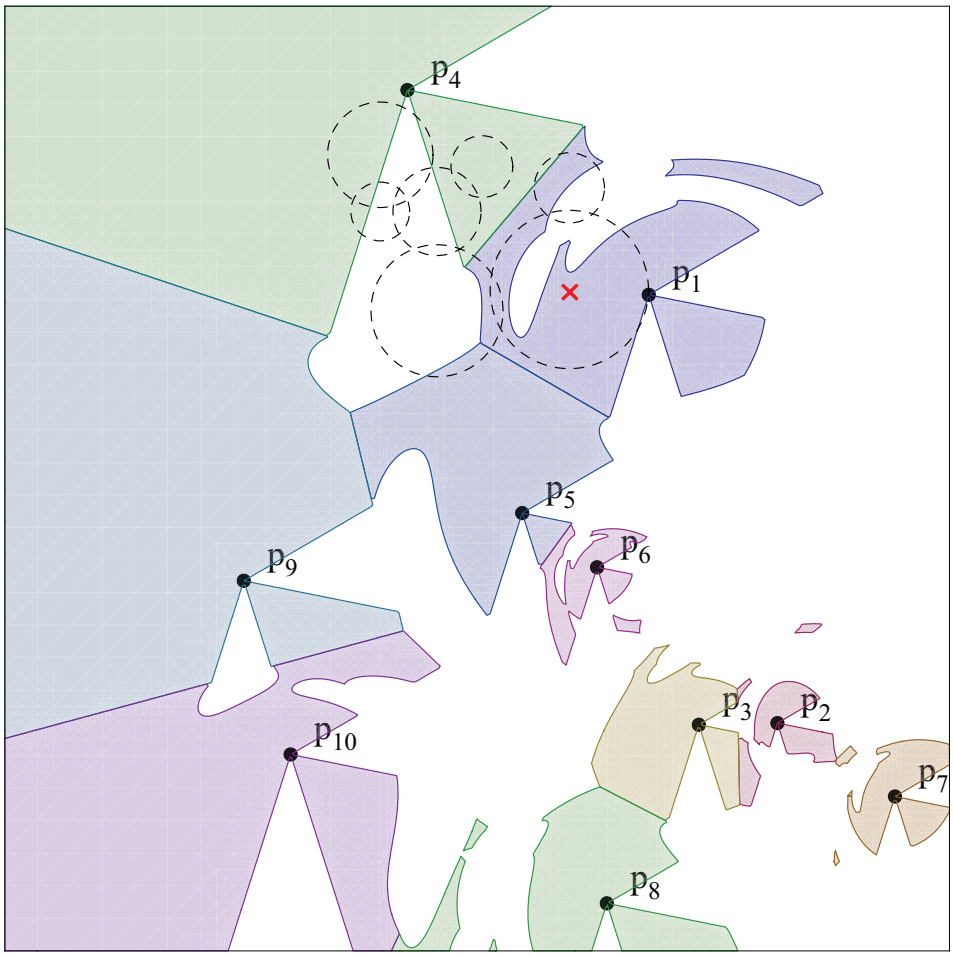


(b) *régions de VPUD et vérification du prédicat d'union de disques vide*

FIGURE 3.8 – *Régions de Voronoï Paramétrée par une Union de Disques (exemple 1). À chaque germe est associée une région de couleur différente*

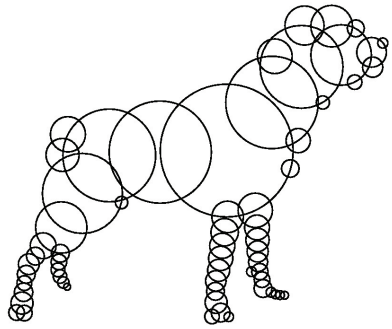


(a) *ensemble de disques considéré*

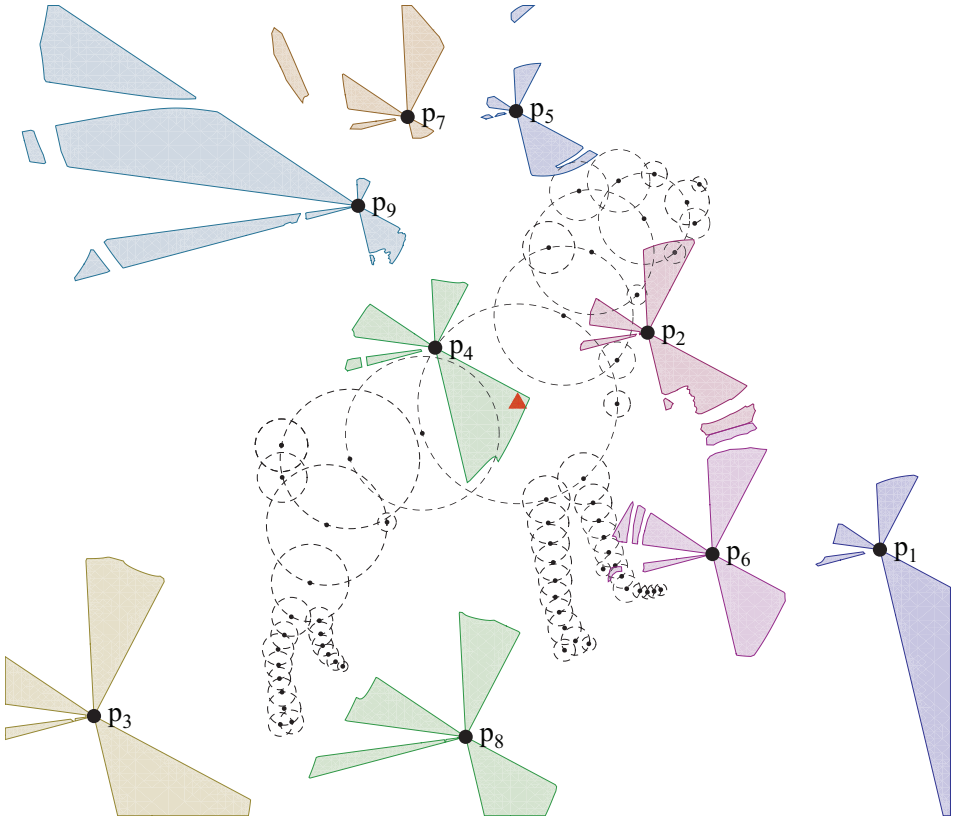


(b) *régions de VPUD et vérification du prédicat d'union de disques vide*

FIGURE 3.9 – *Régions de Voronoï Paramétrée par une Union de Disques (exemple 2). À chaque germe est associée une région de couleur différente*



(a) *ensemble de disques considéré*



(b) *régions de VPUD et vérification du prédicat d'union de disques vide*

FIGURE 3.10 – *Régions de Voronoï Paramétrée par une Union de Disques (exemple 3). À chaque germe est associée une région de couleur différente*

L'union d'ensembles vides est égale à l'ensemble vide. La région $\cup B(p_i, \|p_i - p_i\|, 0)$ représente donc l'ensemble vide. L'intersection entre cette dernière et P est nécessairement vide. Aucune hypothèse n'ayant été faite ni sur P, B ni sur le choix de p_i, ce raisonnement démontre qu'une région de VPUD contient toujours le germe auquel elle est associée. Elle est donc *a fortiori* non vide.

Conclusion du chapitre

Dans ce chapitre, nous avons introduit une nouvelle généralisation du diagramme de Voronoï. Étant donné un nuage de points P, notre diagramme généralisé a l'avantage de regrouper et de structurer en régions tout un ensemble de copies *i.e. formes géométriques similaires à une union de disques donnée* que l'on sait vides, c'est-à-dire ne contenant aucun point de P en leur intérieur.

De plus, cette propriété fondamentale de notre diagramme reste compréhensible car lesdites régions sont représentables en deux dimensions, de même que le diagramme de Voronoï classique.

Pour un nuage de n points, notre diagramme est constitué de n régions, chacune d'entre elles étant paramétrée par une même union de disques $\cup B$. Chacune des régions ainsi définies étend les propriétés des régions de Voronoï classiques à une forme géométrique pouvant être bien plus complexe que le disque.

La définition que l'on vient de donner pour ce nouveau diagramme étant abstraite, nous avons cherché dans un deuxième temps une définition constructive (calculable). Dans la section suivante, nous donnons une caractérisation algébrique des régions et mettons en évidence plusieurs propriétés. L'étude de ces propriétés a été une étape décisive pour représenter les nouvelles régions sur machine.

4 Construction de régions de Voronoï paramétrées par une union de disques

Sommaire

4.1 Caractérisation algébrique d'une région de VPUD . . . 51
4.2 Frontière de la région de VPUD 55
4.2.1 Rappels sur l'étude de courbes planes 55
4.2.2 Propriétés de la frontière d'une région de VPUD 56
4.2.3 Équation caractéristique de la frontière de sur-région élémentaire 57
4.2.4 Inclusion de la courbe-frontière dans une quartique 58
4.3 Propriétés globales des régions de VPUD 61
4.3.1 Inclusion de régions de VPUD aux paramètres similaires . 61
4.3.2 Angles interdits 62
4.3.3 Non partitionnement du plan par régions de VPUD 64
4.3.4 Non connexité d'une région de VPUD 65
4.4 Algorithmes calculant le diagramme de VPUD 65
4.4.1 Difficultés inhérentes à la représentation sur machine . . . 67
4.4.2 Approximation discrète basée sur des points isolés 68
4.4.3 Approximation intérieure basée sur des pavés 79
Conclusion du chapitre 85

Où nous donnons une définition constructive des régions de VPUD, basée sur un système d'inéquations. Ledit système suffit à caractériser algébriquement toute région de VPUD.

Nous étudions les conséquences de cette redéfinition plus précise. Nous avons dans un premier temps étudié la frontière des régions, la constitution et les caractéristiques de cette frontière en tant que courbe plane. On fait appel à ce niveau de résultats classiques de géométrie analytique. Nous avons dans un second temps mis en évidence les propriétés générales et topologiques de ces régions.

À partir de ces nouvelles connaissances, nous avons conçu des moyens pour représenter ces régions sur machine, les calculer et les visualiser en pratique. Nous décrivons tous ces résultats en les illustrant. Nous donnons aussi quelques éléments concernant une technique d'analyse numérique connue : l'arithmétique des intervalles, qui s'est avérée utile dans notre contexte.

4.1 Caractérisation algébrique d'une région de VPUD

Dans cette section, nous montrons que toute région de VPUD peut être construite par intersection ensembliste, à partir d'un certain nombre de régions plus simplement définies. Cette construction est réalisable en deux étapes. Nous présentons le résultat de chaque étape sous la forme d'une propriété. La conjonction des deux propriétés permet finalement de caractériser algébriquement une région de VPUD (comme on le verra au théorème 4.1.6). Caractériser algébriquement signifie trouver un système d'inéquations algébriques, dont l'ensemble de solutions est égal à la définition abstraite.

La définition 3.1.1 de région de Voronoï classique a été rappelée au cours du chapitre précédent. Toute région de Voronoï peut être exprimée comme l'intersection de demi-plans : étant donnés P un ensemble de n points et p_i un germe, la région de Voronoï $V(p_i)$ se décompose en l'intersection de $n-1$ demi-plans. Formellement, la définition :

$$V(p_i) := \left\{x \in \mathbb{R}^2 \mid \|x - p_i\| \leq \left\|x - p_j\right\|, 1 \leq j \leq n\right\}$$

est ainsi équivalente à :

$$V(p_i) = \bigcap_{1 \leq j \leq n} \mathcal{H}\left(p_i, p_j\right)$$

$$\text{où } \mathcal{H}\left(p_i, p_j\right) := \left\{x \in \mathbb{R}^2 \mid \|p_i - x\| \leq \left\|p_j - x\right\|\right\}$$

Chaque demi-plan est une contrainte de distance entre p_i le germe choisi, et p_j un site de P différent de p_i. Algébriquement, cette contrainte de distance se traduit par une inégalité linéaire.

De manière analogue, nous avons établi que les régions de VPUD disposent du même type de propriété. Ainsi, de même que toute région de Voronoï peut être construite par intersection de $n-1$ demi-plans, toute région de VPUD peut être construite par intersection de n régions plus simples. Dans toutes les définitions et propriétés qui suivent, $P = \{p_1, \ldots, p_n\}$ désigne un nuage de points et B un EDS.

Définition 4.1.1 (sur-région). Nous appelons *sur-région* et notons $S_B(p_i, p_j)$ l'ensemble des points du plan, défini pour tout couple de points $p_i, p_j \in P$ tel que :

$$S_B(p_i, p_j) := \left\{x \in \mathbb{R}^2 \mid p_j \notin \cup B(x, \|x - p_i\|, 0)\right\}$$

Propriété 4.1.2. *La région de VPUD associée à B et au germe p_i, notée $V_B(p_i)$, est égale à l'intersection des n sur-régions $S_B(p_i, p_j)$ avec $1 \leq j \leq n$. Formellement cela se traduit par :*

$$V_B(p_i) = \bigcap_{p_j \in P} S_B(p_i, p_j)$$

Démonstration. Pour démontrer la propriété, on part du second membre de l'égalité :

$$x \in \bigcap_{p_j \in P} S_B(p_i, p_j) \Leftrightarrow \forall p_j; p_j \notin \cup B(x, \|x - p_i\|, 0) \text{ par définition de } S_B(p_i, p_j)$$

$$\Leftrightarrow \cup B(x, \|x - p_i\|, 0) \cap P = \emptyset$$

Par définition des régions de VPUD, on en déduit que $x \in V_B(p_i)$. □

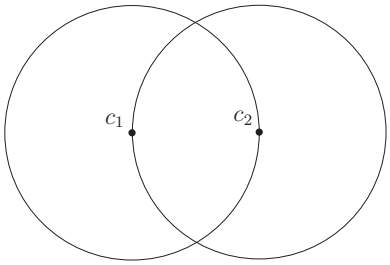


FIGURE 4.1 – *Ensemble de Disques Standard (EDS) utilisé dans les figures suivantes*

On peut construire une région de VPUD par intersection de multiples sur-régions. Et de manière analogue, une sur-région elle-même peut-être construite par intersection ensembliste comme nous le montrons ci-dessous.

Définition 4.1.3 (sur-région élémentaire). Soient deux germes p_i, p_j et un disque particulier $b_k \in B$. Nous appelons *sur-région élémentaire* et notons $S_{b_k}(p_i, p_j)$ l'ensemble de points du plan tel que :

$$S_{b_k}(p_i, p_j) := \left\{ x \in \mathbb{R}^2 \mid p_j \notin b_k \left(x, \|x - p_i\|, 0 \right) \right\}$$

Propriété 4.1.4. *Soit p_i, p_j un couple de points de P. La sur-région $S_B(p_i, p_j)$ associée est égale à l'intersection d'au plus m sur-régions élémentaires $S_{b_k}(p_i, p_j)$. Formellement :*

$$S_B(p_i, p_j) = \bigcap_{b_k \in B} S_{b_k}(p_i, p_j) \text{ quelque soient } p_i, p_j \in P$$

Démonstration. Pour démontrer la propriété, on procède comme précédemment en partant du deuxième membre de l'égalité :

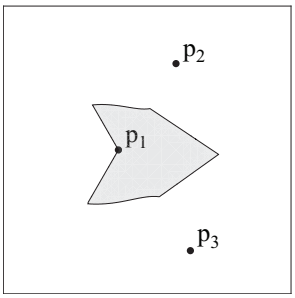
$$\begin{aligned} x \in \bigcap_{b_k \in B} S_{b_k}(p_i, p_j) &\Leftrightarrow \forall b_k \in B, x \in S_{b_k}(p_i, p_j) \\ &\Leftrightarrow \forall b_k \in B, p_j \notin b_k(x, \|x - p_i\|, 0) \text{ par déf. de } S_{b_k}(p_i, p_j) \\ &\Leftrightarrow p_j \notin \cup B(x, \|x - p_i\|, 0) \end{aligned}$$

Par définition des sur-régions, on en déduit que $S_B(p_i, p_j)$. □

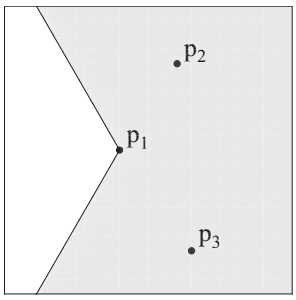
Pour représenter graphiquement ces deux niveaux de décomposition, on a utilisé un EDS simple, comme montré sur la figure 4.1. La figure 4.2 illustre la construction possible d'une région de VPUD à partir de sur-régions. La figure 4.3 illustre la construction possible d'une sur-région à partir de sur-régions élémentaires.

Propriété 4.1.5. *En notant $a = (x, \|x - p_i\|, 0), x \in \mathbb{R}^2$ les configurations considérées par le prédicat d'union de disques vide (prédicat donné dans la définition 3.3.2), l'ensemble des points d'une sur-région est égal à l'ensemble des points vérifiant une inéquation de type :*

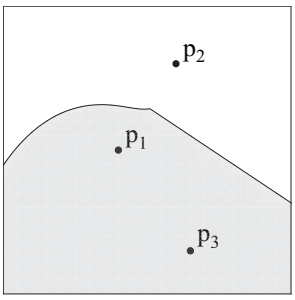
$$\|c_k(a) - p_j\| \geq r_k(a)$$
$$\text{avec } c_k(a) = x + \|x - p_i\| c_k \text{ et } r_k(a) = \|x - p_i\| r_k$$



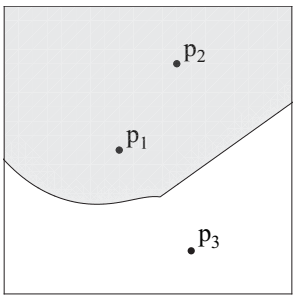
(a) *région de VPUD* $V_B(p_1)$



(b) *sur-région* $S_B(p_1, p_1)$

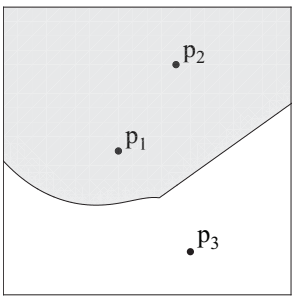


(c) *sur-région* $S_B(p_1, p_2)$

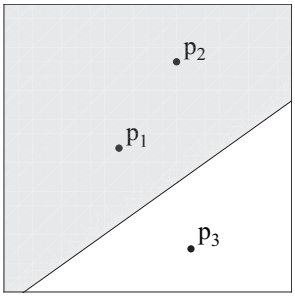


(d) *sur-région* $S_B(p_1, p_3)$

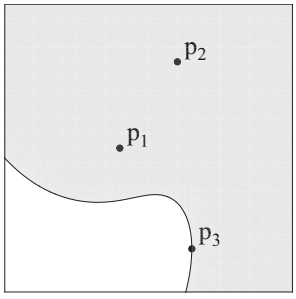
FIGURE 4.2 – *Décomposition d'une région de VPUD en sur-régions. L'intersection ensembliste des trois sur-régions représentées par (b,c,d) est égale à la région de VPUD représentée par (a). L'intérieur d'une région est représenté en gris*



(a) *sur-région* $S_B(p_1, p_3)$



(b) *sur-région élémentaire* $S_{b_1}(p_1, p_3)$



(c) *sur-région élémentaire* $S_{b_2}(p_1, p_3)$

FIGURE 4.3 – *Décomposition d'une sur-région en sur-régions élémentaires. L'intersection ensembliste des deux sur-régions élémentaires représentées par (b,c) est égale à la sur-région représentée par (a). L'intérieur d'une région est représenté en gris*

L'expression de $c_k(a)$ et de $r_k(a)$ provient de la définition 2.2.4 des copies $\cup B(a)$. Les opérations arithmétiques appliquées proviennent de la définition analytique d'un disque.

La précédente remarque est très importante dans la mesure où, grâce à elle et aux deux propriétés 4.1.2, 4.1.4, nous avons pu caractériser algébriquement toute région de VPUD. Ce résultat est décrit dans le théorème ci-dessous. Il signifie une égalité entre région de VPUD, et l'ensemble de solutions d'un certain système algébrique.

Théorème 4.1.6 (caractérisation algébrique). *La région de VPUD $V_B(p_i)$ est égale à l'ensemble des solutions du système algébrique suivant :*

$$\begin{cases} \|c_1(a) - p_1\| \geq r_1(a) \\ \vdots \\ \|c_k(a) - p_j\| \geq r_k(a) \\ \vdots \\ \|c_m(a) - p_n\| \geq r_m(a) \end{cases}$$

avec $c_1, c_2, \ldots, c_m$ et $r_1, r_2, \ldots, r_m$ les centres et rayons respectifs des disques de B, et avec $a = (x, \|x - p_i\|, 0)$, $x \in \mathbb{R}^2$ l'ensemble des configurations dépendant de p_i qui sont considérées par la définition 3.3.3 des régions de VPUD. Les coordonnées du point x constituent les deux variables réelles du système algébrique.

Démonstration. D'une part, la propriété 4.1.2 a montré qu'une région de VPUD peut s'obtenir par intersection d'au plus n sur-régions $S_B(p_i, p_j)$, $p_j \in P$. D'autre part, la propriété 4.1.4 a montré qu'une sur-région peut elle même s'obtenir par intersection d'au plus m sur-régions élémentaires $S_{b_k}(p_i, p_j)$, $b_k \in B$. Donc :

(A) une région de VPUD s'obtient en intersectant au plus $m \times n$ sur-régions élémentaires.

(B) il a été dit que toute sur-région élémentaire correspond à l'ensemble des solutions d'une inéquation algébrique de la forme :

$$\|c_k(a) - p_j\| \geq r_k(a)$$

La conjonction de ces deux faits (A,B) suffit pour conclure. □

Remarque. En notant n le cardinal de l'ensemble P et m le cardinal de l'ensemble B, le système algébrique équivalent à la définition d'une région de VPUD est constitué d'au plus $(m \times n) - 1$ inéquations. En effet : d'une part tout couple p_j, b_k engendre une sur-région élémentaire, modélisable par une inéquation. D'autre part, une de ces inéquations peut être supprimée.

Vérifions-le. Dans le théorème précédent, comme dans la définition 3.3.3 des régions de VPUD, les configurations considérées pour les copies $\cup B(a)$ sont toutes de type : $a = (x, \|x - p_i\|, 0)$ avec $x \in \mathbb{R}^2$. En considérant le cas particulier $p_j = p_i$ et $b_k = b_1$, l'inéquation $\|c_k(a) - p_j\| \geq r_k(a)$ devient $\|x - p_i\| \geq \|x - p_i\|$, une inéquation qui est toujours vraie. Celle-ci n'influençant en rien la solution du système d'inéquations, elle est inutile.

Dans cette section nous avons caractérisé algébriquement les points faisant partie d'une région de VPUD.

Cette caractérisation algébrique nous a permis de redéfinir les régions de VPUD d'une façon constructive, en partant de régions du plan délimitées par une unique courbe algébrique, et en procédant par intersection ensembliste. On dispose ainsi d'un moyen algorithmique pour vérifier l'appartenance de tout point du plan à une région. La représentation de la région dans sa totalité reste toutefois problématique. Nous avons étudié la frontière des régions de VPUD pour y remédier, la section suivante détaille les résultats obtenus.

4.2 Frontière de la région de VPUD

En informatique, on a coutume de représenter les régions bornées du plan d'après leur frontière. Pour les régions de Voronoï classique par exemple, on utilise des polygones, c'est-à-dire une liste de points ordonnés dans le « bon » ordre.

Notre but final étant la représentation des régions de VPUD nouvellement définies sur machine, nous avons commencé par étudier en détail la frontière de celles-ci. La présente section expose les résultats obtenus.

Le théorème 4.1.6 énonce l'équivalence entre : région de VPUD et solutions d'un système algébrique. Chacune des inéquations composant ledit système correspond à une contrainte de distance. La remarque est tout aussi valable pour le diagramme de Voronoï classique. Par contre, dans notre cas, certaines contraintes de distance sont bien plus complexes.

Étant donné que chaque disque b_k de B possède une position et une taille de base, que ces deux données sont ensuite paramétrées par le point x considéré (voir la définition 3.3.3 des régions de VPUD), il vient qu'une partie des inéquations modélisant les contraintes de distance sont *non linéaires*. Cette section présente quelques rappels sur l'étude de courbes planes, on présente ensuite le caractère *composé* de la frontière d'une région de VPUD et les courbes constituant les frontières.

4.2.1 Rappels sur l'étude de courbes planes

L'étude des courbes planes est une branche ancienne de la géométrie analytique, ses résultats remontent à l'époque de Descartes et Leibniz. Les courbes planes sont un sujet plus compliqué qu'il n'y parait à première vue. L'article de Husemöller (1987) concernant la famille des courbes dites elliptiques, montre que des courbes relativement simples *a priori* renferment une complexité mathématique remarquable et constituent un champs de recherche à part entière.

Nous rappelons ici quelques définitions utiles à nos propres travaux. Une *courbe plane* est définie comme étant un ensemble de points contenus dans un unique plan. Une courbe plane est identifiable via une fonction caractéristique : $\alpha : I \to \mathbb{R}^2$, où I est un intervalle réel c'est-à-dire un sous-ensemble continu de $\mathbb{R}$. Une autre représentation possible est l'équation cartésienne implicite : $f(x, y) = 0$. Il existe de multiples classifications

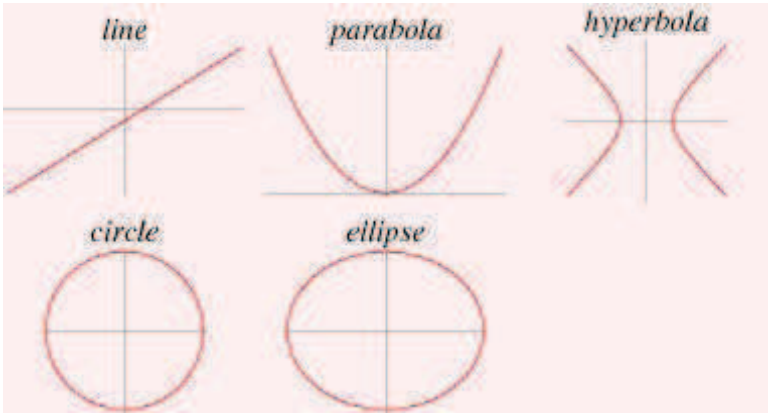


FIGURE 4.4 – *Exemples de courbes planes parmi les plus connues.*

pour les courbes planes, décrites dans divers ouvrages, voir les travaux de Nicaise (2011), Jean-Marie Arnaudiès (1996).

Une courbe peut-être soit fermée soit ouverte. Des exemples de courbes ouvertes sont la *droite*, la *parabole*, l'*hyperbole*, *etc.* Des exemples de courbes fermées sont le *cercle* et l'*ellipse*, *etc.* Nous représentons graphiquement ces exemples élémentaires figure 4.4.

On dit qu'une courbe plane est une courbe plane *algébrique* lorsque son équation cartésienne $f(x, y) = 0$ est de forme polynomiale. Le degré d'une courbe plane algébrique est le degré maximum des monômes en x et y du polynôme $f(x, y)$. Voyons quelques exemples :

- les courbes algébriques de degré 1 sont des droites
- les courbes algébriques de degré 2 sont appelées coniques. Des exemples familiers sont : le cercle, l'ellipse, *etc.*
- une courbe algébrique de degré 3 est appelée une cubique
- une courbe algébrique de degré 4 est appelée une quartique

4.2.2 Propriétés de la frontière d'une région de VPUD

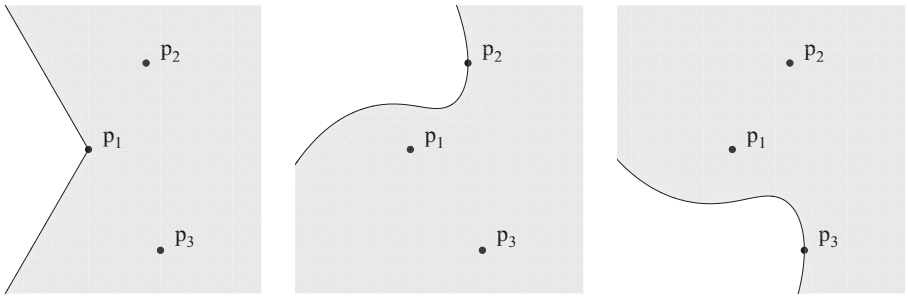
Rappelons qu'une région de VPUD peut se décomposer en sur-régions puis en sur-régions élémentaires. Ceci a une conséquence directe sur la nature de sa frontière : celle-ci également est décomposable en morceaux de courbe. La frontière contient des points remarquables, qui font la « jonction », comme expliqué ci-dessous.

Définition 4.2.1 (point de jonction). Soit $p_i \in P$ un germe. On appelle *point de jonction* tout point de la frontière $\partial V_B(p_i)$ qui appartient simultanément aux frontières respectives $\partial S_{b_k}(p_i, p_j), \partial S_{b_{k'}}(p_i, p_{j'})$ de deux sur-régions élémentaires distinctes (c'est-à-dire que $k \neq k'$ ou $j \neq j'$). Formellement, un point de la frontière d'une région $V_B(p_i)$ de Voronoï paramétrée par une union de disques est point de jonction, si et seulement si :

$$\exists (b_k, p_j), (b_{k'}, p_{j'}) \in B \times P \text{ tel que } (b_k, p_j) \neq (b_{k'}, p_{j'});$$
$$x \in \partial S_{b_k}(p_i, p_j) \text{ et } x \in \partial S_{b_{k'}}(p_i, p_{j'})$$

Propriété 4.2.2 (finitude des points de jonction). *Soient P un nuage de points et B un EDS. Pour toute région de VPUD notée $V_B(p_i)$, les points de jonction existants sur sa frontière sont en nombre fini.*

La preuve de cette propriété est fournie en Annexe A.1.



(a) (b) (c)

FIGURE 4.5 – *Non-linéarité de la frontière des sur-régions élémentaires, dans le cas général*

4.2.3 Équation caractéristique de la frontière de sur-région élémentaire

La notion de sur-région élémentaire a été introduite définition 4.1.3. La propriété 4.1.5 a énoncé le fait qu'une telle sur-région élémentaire $S_{b_k}(p_i, p_j)$ pouvait être caractérisée par une inégalité du type :

$$\|c_k(a) - p_i\| \geq r_k(a)$$

où a est une configuration du type : $(x, \|x - p_i\|, 0), x \in \mathbb{R}^2$. La frontière d'une sur-région élémentaire se caractérise alors par une équation semblable, que nous avons développé davantage au cours de nos travaux.

Dans cette sous-section, on présente les résultats concernant ce développement. Ainsi, on décrit plus précisément les courbes-frontières d'une sur-région élémentaire (voir les illustrations de la figure 4.5).

Propriété 4.2.3. *Étant donné un disque $b_k \in B$ et un couple de points $p_i, p_j \in P$, la frontière d'une sur-région élémentaire $S_{b_k}(p_i, p_j)$, est décrite par une équation cartésienne implicite de la forme :*

$$(\mathcal{C}) : \left(X^2 + Y^2\right) \frac{\gamma^2 + \delta^2 + \epsilon}{2} + \sqrt{X^2 + Y^2}\left(\gamma X + \delta Y + \alpha\gamma + \beta\delta\right) + \left(\alpha X + \beta Y + \frac{\alpha^2 + \beta^2}{2}\right) = 0 \quad (4.1)$$

Où $\alpha, \beta, \gamma, \delta, \epsilon$ sont des constantes calculées à partir des données p_i, p_j, c_k, r_k. Plus précisément :

$$\begin{array}{ll} (\alpha, \beta) & = p_i - p_j \\ (\gamma, \delta) & = c_k \\ \epsilon & = 1 - r_k^2 \end{array}$$

Démonstration. D'après la propriété 4.1.4 on sait que l'inéquation $\left\|c_k(a) - p_j\right\| \geq r_k(a)$ avec $a = (x, \|x - p_i\|, 0)$ modélise l'ensemble des points d'une sur-région élémentaire $S_{b_k}(p_i, p_j)$. En remplaçant $c_k(a)$ et $r_k(a)$ par leurs valeurs respectives, cette expression devient alors :

$$\left\|F(x) - p_j\right\| \geq \|x - p_i\| r_k \text{ avec } F(x) = x + \|x - p_i\|$$

Où c_k, r_k sont des constantes dépendantes de B et p_i, p_j des germes. Les deux termes de l'égalité étant positifs, on peut élever au carré sans changer les solutions de l'équation. Il vient que :

$$\left\|F(x) - p_j\right\|^2 - \|x - p_i\|^2 r_k^2 = 0$$

On applique le changement de variable : $t = (X, Y) = x - p_i$. Il vient que :

$$\left\| p_i + t + \|t\| \, c_k - p_j \right\|^2 - \|t\|^2 r_k^2 = 0$$

La norme peut être décomposée en employant le produit scalaire. On note $\langle a; b \rangle$ le produit scalaire de deux vecteurs a, b.

$$\left\| p_i - p_j \right\|^2 + \|t + \|t\| \, c_k\|^2 + 2\langle p_i - p_j; t + \|t\| \, c_k \rangle - \|t\|^2 r_k^2 = 0 \qquad (4.2)$$

Développons le second terme de la somme présente dans cette dernière équation.

$$\|t + \|t\| \, c_k\|^2 = \|t\|^2 \|c_k\|^2 + 2 \|t\| \, \langle t; c_k \rangle$$

Développons le troisième terme de cette même somme dans l'équation 4.2.

$$\langle p_i - p_j; t + \|t\| \, c_k \rangle = \langle p_i - p_j; t \rangle + \|t\| \, \langle p_i - p_j; c_k \rangle$$

Il vient alors que l'équation 4.2 est équivalente à :

$$\left\| p_i - p_j \right\|^2 + \Big(\, \|t\|^2 \|c_k\|^2 + 2 \|t\| \, \langle t; c_k \rangle \Big) \; + \\ 2 \left(\langle p_i - p_j; t \rangle + \|t\| \, \langle p_i - p_j; c_k \rangle \right) - \|t\|^2 r_k^2 = 0$$

On factorise le terme $\|t\|$:

$$\|t\|^2 \left(1 + \|c_k\|^2 - r_k^2 \right) + 2 \|t\| \left(\langle t; c_k \rangle + \langle p_i - p_j; c_k \rangle \right) + \\ 2\langle t; p_i - p_j \rangle + \left\| p_i - p_j \right\|^2 = 0$$

On peut diviser par 2 sans changer le sens de l'équation :

$$(\mathcal{C}') : \|t\|^2 \left(\frac{1 + \|c_k\|^2 - r_k^2}{2} \right) + \|t\| \, \langle c_k; p_i - p_j + t \rangle + \\ \langle t; p_i - p_j \rangle + \frac{\left\| p_i - p_j \right\|^2}{2} = 0$$

De cette ligne, en procédant par identification avec les réels $\alpha, \beta, \gamma, \delta, \epsilon$ comme définis dans l'énoncé de la propriété, on conclut. □

Durant nos recherches, nous avons observé que cette équation implicite non-linéaire peut sous certaines conditions se simplifier grandement. Ces *cas particuliers* donnent parfois lieu à des courbes plus « simples », linéaires ou linéaires par morceaux. Sans prétendre à l'exhaustivité, nous avons recensé deux tels cas. Nous décrivons ceux-ci en annexe A.2.

4.2.4 Inclusion de la courbe-frontière dans une quartique

Les propriétés vues jusqu'ici permettent d'affirmer que la frontière d'une sur-région élémentaire *n'est pas* une courbe algébrique (son équation caractéristique contient des racines carrées). Durant notre étude, nous avons montré qu'une telle courbe-frontière fait néanmoins partie d'une autre courbe, qui est une courbe algébrique quant à elle. Ce résultat est démontré formellement ci-dessous.

Dans la classe des courbes algébriques, il existes des classes plus fines. La notion de *degré d'une courbe algébrique* a déjà été rappelée à la sous-section 4.2.1. Ci-dessous, nous citons un passage du site *Web* de Mehl (2009 (accédé le 5 mars 2012), qui décrit la terminologie existante. Nous nous servons de cette terminologie pour décrire le plus précisément possible la nature des courbes engendrées par nos définitions.

« Une importante catégorie de courbes algébriques est celle dont le terme de plus haut degré contient en facteur $x^2 + y^2$, c'est-à-dire le carré de : la distance de l'origine $\mathcal{O}$ à un point de la courbe. On rencontre fréquemment ces courbes dans des lieux géométriques faisant intervenir des cercles. Elles sont qualifiées de circulaires si $x^2 + y^2$ intervient au 1er degré, voire bicirculaires si $x^2 + y^2$ intervient au carré. Dans cette catégorie, on rencontre couramment les cubiques circulaires et les quartiques bicirculaires ».

Propriété 4.2.4. *Toute courbe du type $\mathcal{C}$ présentée au lemme 4.2.3 fait partie d'une courbe algébrique dite bicirculaire selon la classification standard des courbes planes :*

$$\begin{aligned}\mathcal{C}' : &- (ac + bd + cX + dY)^2 \left(X^2 + Y^2\right) \\ &+ \left(\frac{1}{2}\left(a^2 + b^2\right) + aX + bY + \frac{1}{2}\left(c^2 + d^2 + e\right)\left(X^2 + Y^2\right)\right)^2 = 0\end{aligned}$$

Démonstration. Un point du plan, de coordonnées (X, Y) appartient à une courbe de type $(\mathcal{C})$ si et seulement si :

$$\begin{aligned}&\left(X^2 + Y^2\right)\frac{c^2 + d^2 + e}{2} \\ &\qquad + \sqrt{X^2 + Y^2}\,(cX + dY + ac + bd) + \left(aX + bY + \frac{a^2 + b^2}{2}\right) = 0\end{aligned}$$

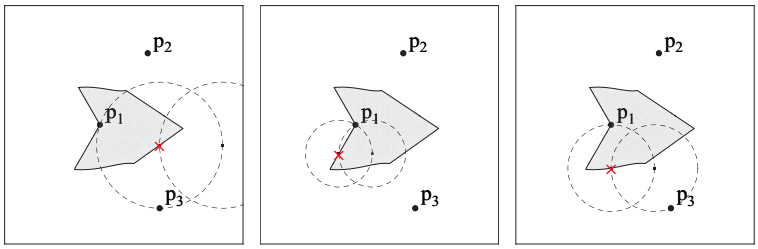
Cette équation est équivalente à :

$$\begin{aligned}\mathcal{C} : &\left(X^2 + Y^2\right)\frac{c^2 + d^2 + e}{2} \\ &+ \left(aX + bY + \frac{a^2 + b^2}{2}\right) = -\sqrt{X^2 + Y^2}\,(cX + dY + ac + bd)\end{aligned}$$

Si un point de coordonnées (x, y) satisfait cette équation, cela implique que le même point satisfait *aussi* l'équation que l'on obtient par élévation des deux termes à la puissance deux. Cette dernière équation s'écrit :

$$\begin{aligned}\mathcal{C}' : &\left[\left(X^2 + Y^2\right)\frac{c^2 + d^2 + e}{2} + aX + bY + \frac{a^2 + b^2}{2}\right]^2 \\ &- \left(X^2 + Y^2\right)(cX + dY + ac + bd)^2 = 0\end{aligned}$$

Ainsi, l'appartenance du point de coordonnées (X, Y) à une courbe type $\mathcal{C}$ implique l'appartenance à une courbe type $\mathcal{C}'$. Autrement dit, toute courbe type $\mathcal{C}$ est un sous-ensemble d'une courbe type $\mathcal{C}'$. La racine carrée a été supprimée, l'équation cartésienne d'une courbe type $(\mathcal{C}')$ est composée d'opérations arithmétiques uniquement ; une courbe type $(\mathcal{C}')$ est est une courbe algébrique. De plus, comme une telle équation possède $\left(X^2 + Y^2\right)$ comme terme de plus haut degré, une courbe de type $\mathcal{C}'$ est une courbe algébrique et bicirculaire. □



(a) (b) (c)

FIGURE 4.6 – *Caractère non polygonal d'une région de VPUD.*

Propriété 4.2.5 (nature quartique des courbes type C'). *Toute courbe du type de C' présentée plus tôt est une courbe algébrique de degré 4, autrement dit : une quartique. Elle est décrite par une équation de la forme :*

$$\begin{aligned} C' : Ax^4 + By^4 + Cx^3y + Dx^2y^2 + Exy^3 + Fx^3 + Gy^3 + Hx^2y \\ + Ixy^2 + Jx^2 + Ky^2 + Lxy + Mx + Ny + O = 0 \end{aligned}$$

Avec :

$$\begin{aligned} A &= -4\gamma^2 + \gamma^4 + 2\gamma^2\delta^2 + \delta^4 + 2\gamma^2\epsilon + 2\delta^2\epsilon + \epsilon^2 \\ B &= \gamma^4 - 4\delta^2 + 2\gamma^2\delta^2 + \delta^4 + 2\gamma^2\epsilon + 2\delta^2\epsilon + \epsilon^2 \\ C &= -8\gamma\delta \\ D &= -4\gamma^2 + 2\gamma^4 - 4\delta^2 + 4\gamma^2\delta^2 + 2\delta^4 + 4\gamma^2\epsilon + 4\delta^2\epsilon + 2\epsilon^2 \\ E &= -8\gamma\delta \\ F &= -4\alpha\gamma^2 - 8\beta\gamma\delta + 4\alpha\delta^2 + 4\alpha\epsilon \\ G &= 4\beta\gamma^2 - 8\alpha\gamma\delta - 4\beta\delta^2 + 4\beta\epsilon \\ H &= 4\beta\gamma^2 - 8\alpha\gamma\delta - 4\beta\delta^2 + 4\beta\epsilon \\ I &= -4\alpha\gamma^2 - 8\beta\gamma\delta + 4\alpha\delta^2 + 4\alpha\epsilon \\ J &= 4\alpha^2 - 2\alpha^2\gamma^2 + 2\beta^2\gamma^2 - 8\alpha\beta\gamma\delta + 2\alpha^2\delta^2 - 2\beta^2\delta^2 + 2\alpha^2\epsilon + 2\beta^2\epsilon \\ K &= 4\beta^2 - 2\alpha^2\gamma^2 + 2\beta^2\gamma^2 - 8\alpha\beta\gamma\delta + 2\alpha^2\delta^2 - 2\beta^2\delta^2 + 2\alpha^2\epsilon + 2\beta^2\epsilon \\ L &= 8\alpha\beta \\ M &= 4\alpha^3 + 4\alpha\beta^2 \\ N &= 4\alpha^2\beta + 4\beta^3 \\ O &= \alpha^4 + 2\alpha^2\beta^2 + \beta^4 \end{aligned}$$

Où $a, \beta, \gamma, \delta, \epsilon$ sont les coefficients réels définis au lemme 4.2.3.

Du fait de la longueur des calculs, la démonstration de ce résultat est donnée en annexe A.3.

La figure 4.6 représente une approximation fine d'une région de VPUD. On observe bien que la frontière est composée en partie de segments de droites, en partie de morceaux de courbes.

Notre étude de la frontière d'une sur-région élémentaire a mis en évidence le caractère non-linéaire, *a fortiori* non-polygonal de cette frontière. Les caractéristiques sont les mêmes pour la frontière d'une région de

VPUD. Ceci constitue une vraie difficulté pour représenter sur machine des régions de VPUD et pouvoir les utiliser pour réaliser des calculs.

Pour acquérir plus de connaissances avant la conception d'algorithmes pouvant calculer les régions de VPUD, nous avons également étudié les propriétés qui concernent la région *globalement*, et pas seulement sa frontière. La section suivante présente donc les propriétés concernant l'intérieur des régions de VPUD.

4.3 Propriétés globales des régions de VPUD

4.3.1 Inclusion de régions de VPUD aux paramètres similaires

Nous nous sommes interrogé sur la relation qui lie une région de Voronoï paramétrée par une union de disques $\cup B$, et une région semblable paramétrée par un sous-ensemble B' de B. Nous présentons dans cette sous-section un résultat démontré à ce sujet.

Propriété 4.3.1 (inclusion dans une région de VPUD aux paramètres similaires). *On admet que pour tout ensemble B de disques standard composé de deux disques ou plus, il existe un sous-ensemble $B' \subsetneq B$ qui soit lui aussi un EDS. Soit B, B' deux ensembles de disques standard distincts tels que B' est un sous-ensemble strict de B. Quelque soit le germe $p_i \in P$ considéré, la région de VPUD associée aux données (B, p_i) est incluse dans la région de VPUD associée aux donnée (B', p_i). Formellement :*

$$B' \text{ est un EDS et } B' \subsetneq B \Rightarrow \forall p_i \in P, V_B(p_i) \subset V_{B'}(p_i)$$

Démonstration. Nous avons admis que pour tout ensemble B de disques standard composé de deux disques ou plus, il existe un sous-ensemble $B' \subsetneq B$ qui soit lui aussi EDS. Éventuellement, B' peut être réduit au singleton du seul disque unité. Un tel singleton est bien un EDS. On remarque que B peut être décomposé en deux sous-ensembles disjoints : $B = B' \cup B''$. Par définition de la région de VPUD nous avons :

$$x \in V_B(p_i) \Leftrightarrow \cup B(x, \|x - p_i\|, 0) \cap P = \emptyset$$

En décomposant B comme expliqué précédemment, l'équation $\cup B(x, \|x - p_i\|, 0) \cap P = \emptyset$ est équivalente à :

$$\left[\left(\cup B'(x, \|x - p_i\|, 0)\right) \cup \left(\cup B''(x, \|x - p_i\|, 0)\right)\right] \cap P = \emptyset$$

Par conséquent :

$$x \in V_B(p_i) \Leftrightarrow \left(\cup B'(x, \|x - p_i\|, 0)\right) \cup P = \emptyset \text{ et } \left(\cup B''(x, \|x - p_i\|, 0)\right) \cap P = \emptyset$$

D'autre part, par définition :

$$\cup B'(x, \|x - p_i\|, 0) \cap P \Leftrightarrow x \in V_{B'}(p_i)$$

De ces deux derniers faits on déduit : $x \in V_B(p_i) \Rightarrow x \in V_{B'}(p_i)$. De là on conclut : $V_{B'}(p_i) \subset V_B(p_i)$ □

Le résultat que nous venons de démontrer est illustré figure 4.7. Une conséquence directe de ce résultat est que toute région de VPUD associée à un germe p_i est sous-ensemble de la région de Voronoï classique associée au même germe p_i, comme énoncé par le corollaire ci-dessous.

Corollaire (inclusion dans une région de Voronoï)**.** *On admet que pour tout ensemble B de disques standard composé de deux disques ou plus, il existe un sous-ensemble $B' \subsetneq B$ qui soit lui aussi EDS. Étant donné B un EDS, pour tout germe p_i, la région de VPUD associée à p_i est incluse dans la région de Voronoï classique associée à p_i. Formellement :*

$$\forall p_i \in P, V_B(p_i) \subset V(p_i)$$

4.3.2 Angles interdits

Propriété 4.3.2 (angles interdits)**.** *Il existe des ensembles de disques standard B pour lesquels, pour tout point $p_i \in P$, il existe un intervalle I inclus dans l'intervalle $[0, 2\pi[$ tel que : pour tout point x du plan, si l'angle formé entre le vecteur $\overrightarrow{p_i x}$ et l'axe des abscisses appartient à l'intervalle I, alors x est nécessairement exclu de la région de VPUD associée à p_i. Formellement :*

$$\forall x \in \mathbb{R}^2, \angle\left(\overrightarrow{p_i X}, \overrightarrow{p_i x}\right) \in I \Rightarrow x \notin V_B(p_i)$$

où $V_B(p_i)$ dénote la région de VPUD associée au germe $p_i \in P$.

Démonstration. Posons $B = \{b_1, b_2\}$ avec $b_1 = b(\mathcal{O}, 1)$ et $b_2 = b((1,0), 1)$. Soit P le nuage de points que l'on se donne, et soit $p_i \in P$ un point en particulier. Soit $x \in \mathbb{R}^2$ un point du plan. D'une part, par définition d'une région de VPUD on a :

$$\begin{aligned} x \in V_B(p_i) &\Leftrightarrow \bigcup B(a) \cap P = \emptyset \text{ où } a = (x, \|x - p_i\|, 0) \\ \text{donc } x \notin V_B(p_i) &\Leftrightarrow \bigcup B(a) \cap P \neq \emptyset \end{aligned}$$

D'autre part, en particulier, il est vrai que :

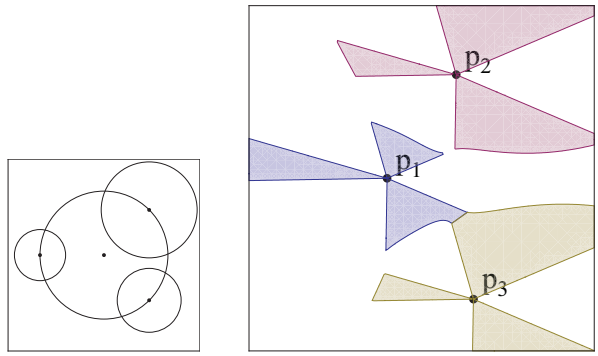
$$p_i \in b_2(a) \Rightarrow \bigcup B(a) \cap P = \emptyset \Rightarrow x \notin V_B(p_i)$$

Or,

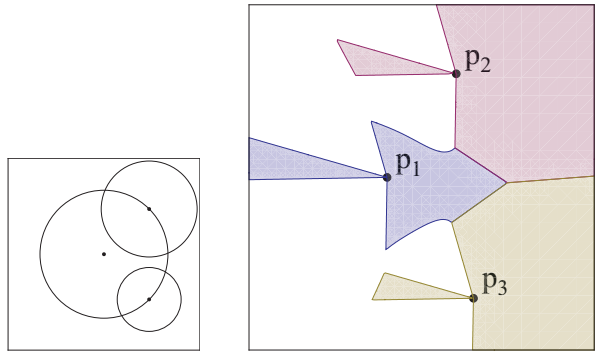
$$\begin{aligned} p_i \in b_2(a) &\Leftrightarrow \|x + \|p_i - x\| c_2 - p_i\| < \|p_i - x\| r_2 \\ &\Leftrightarrow \left(1 + \|c_2\|^2 - r_2^2\right) + 2\langle M_{\angle\left(\overrightarrow{p_i X}, \overrightarrow{p_i x}\right)} \overrightarrow{u}; c_2\rangle < 0 \end{aligned}$$

Où $\overrightarrow{u}$ représente le vecteur unité parallèle à l'axe des abscisses, M_θ une matrice de rotation d'angle θ, ici en l'occurrence, $\theta = \angle\left(\overrightarrow{p_i X}, \overrightarrow{p_i x}\right)$. Ainsi, $\left(1 + \|c_2\|^2 + r_2^2\right) + 2\langle M_{\angle\left(\overrightarrow{p_i X}, \overrightarrow{p_i x}\right)} \overrightarrow{u}; c_2\rangle < 0 \Rightarrow x \notin V_B(p_i)$. Dans notre cas, remplaçons les paramètres c_2, r_2 par leurs valeurs respectives. De cette application numérique, il vient que :

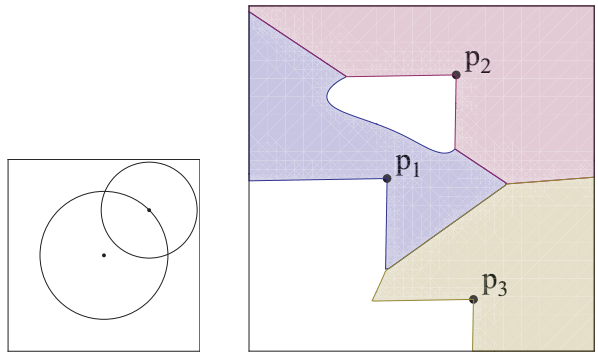
$$1 + 1 - 1 + 2cos(\angle\left(\overrightarrow{p_i X}, \overrightarrow{p_i x}\right)) < 0$$



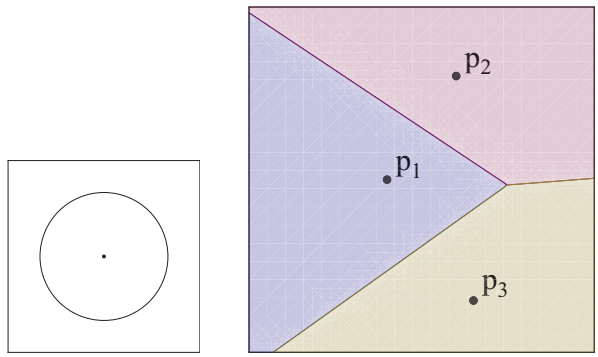
(a) B_4 (b) $V_{B_4}(P)$



(c) B_3 (d) $V_{B_3}(P)$



(e) B_2 (f) $V_{B_2}(P)$



(g) B_1 (h) $V_{B_1}(P)$

FIGURE 4.7 – *Propriété d'inclusion entre régions de Voronoï aux paramètres similaires : une région de VPUD associée à un certain germe et paramétrée par un ensemble de disques B est incluse dans la région de VPUD associée au même germe et paramétrée par un sous-ensemble de B. Le retrait d'un disque de B a pour effet de « remplir » le plan avec de nouveaux emplacements libres*

Ce qui équivaut à :

$$cos(\angle\left(\overrightarrow{p_iX}, \overrightarrow{p_ix}\right)) < -\frac{1}{2} \tag{4.3}$$

À partir des résultats classiques de trigonométrie et d'analyse, on sait que $cos\left(\frac{2}{3}\pi\right) = -\frac{1}{2}$ et $cos\left(\frac{4}{3}\pi\right) = -\frac{1}{2}$, et que la fonction cosinus est strictement décroissante entre $\frac{2}{3}\pi$ et π, strictement croissante entre π et $\frac{4}{3}\pi$. Par conséquent, l'inéquation 4.3 équivaut à :

$$\angle\left(\overrightarrow{p_iX}, \overrightarrow{p_ix}\right) \in \left[\frac{2}{3}\pi, \frac{4}{3}\pi\right[$$

En posant $I = \left[\frac{2}{3}\pi, \frac{4}{3}\pi\right]$ on a bien $I \subset [0, 2\pi[$ et pour tout point du plan $x \in \mathbb{R}^2$, $\angle\left(\overrightarrow{p_iX}, \overrightarrow{p_ix}\right) \in I \Rightarrow x \notin V_B(p_i)$ □

4.3.3 Non partitionnement du plan par régions de VPUD

Le corrolaire 4.3.1 indique que toute région de VPUD associée à un germe p_i est incluse dans la région de Voronoï classique associée au même germe. La propriété 4.3.2 indique qu'il existe des jeux de données pour lesquels on peut savoir à l'avance quels points seront exclus de la région de VPUD. En associant ces deux résultats avec les connaissances des régions de Voronoï classiques, nous pouvons montrer une propriété qui concerne le diagramme de VPUD en totalité, propriété énoncée ci-dessous.

Propriété 4.3.3 (non-partitionnement du plan)**.** *Soit P un nuage de points et B un EDS constitué d'au minimum deux disques. Le diagramme de VPUD noté $V_B(P)$ qui désigne l'ensemble de toutes les régions $V_B(p_i), p_i \in P$, n'est pas une partition du plan. Autrement dit, il existe au moins un point du plan qui n'appartient à aucune région de VPUD, et ce quelque soit son germe p_i. Formellement :*

$$\exists x \in \mathbb{R}^2, \forall p_i \in P, x \notin V_B(p_i)$$

Démonstration. Nous faisons trois observations préliminaires.

1. Toute région de VPUD est incluse dans la région de Voronoï classique associée au même germe.
2. En supposant que l'ensemble B de disques standard est constitué d'au moins deux disques, la propriété des angles interdits peut s'appliquer quelque soit le germe p_i considéré. Il vient que l'inclusion d'une région de VPUD dans une région de Voronoï classique associée est une inclusion stricte.
3. Les régions de Voronoï classiques constituent une partition du plan, donc en particulier les intérieurs respectifs des régions de Voronoï classiques sont deux à deux disjoints.

À partir des observations (2,3) on déduit que pour un EDS d'au moins deux disques noté B, quelque soit le nuage de points P et le point générateur $p_k \in P$ choisi, il existe des points ne se trouvant pas dans la région de VPUD associée à p_i, et n'appartenant pas non plus à toute autre région de VPUD, car appartenant à la région de Voronoï classique associée à p_i.
De ce fait, l'ensemble des régions de VPUD n'est pas une partition du plan. □

4.3.4 Non connexité d'une région de VPUD

Au cours de nos travaux, nous avons montré qu'une région de VPUD n'est pas toujours connexe. Ci-dessous on exhibe un exemple de jeu de données pour lequel la région de VPUD est non connexe.

Propriété 4.3.4 (non-connexité d'une région de VPUD). *Il existe un triplet : nuage de points P, germe p_i, ensemble de disques standard B, tel que la région $V_B(p_i)$ est non-connexe. Formellement :*

$$\exists A_1, A_2 \subset \mathbb{R}^2; A_1 \cap A_2 = \emptyset \; et \; [A_1 \cap V_B(p_i)] \cup [A_2 \cap V_B(p_i)] = V_B(p_i)$$

Les données utilisées pour établir cette propriété sont :

$$\begin{array}{lllllllll} P & = & \{p_1, p_2\} & B & = & \{b(c_1, r_1), b(c_2, r_2)\} & & & \\ p_1 & = & (0,0) & c_1 & = & (0,0) & \text{et } r_1 & = & 1 \\ p_2 & = & (\frac{1}{2}, 2) & c_2 & = & (1,0) & \text{et } r_2 & = & 1 \end{array}$$

Pour ce jeu de données, l'union de disques $\bigcup B$ est l'union de deux disques de rayon 1 et placés sur un même axe horizontal. On obtient une région de VPUD $V_B(p_1)$ qui est séparée en deux composantes disjointes. Les détails de la démonstration sont donnés en annexe A.4.

Un autre exemple de jeu de données, plus élaboré, qui génère également une région de VPUD non connexe est illustré sur la figure 4.8.

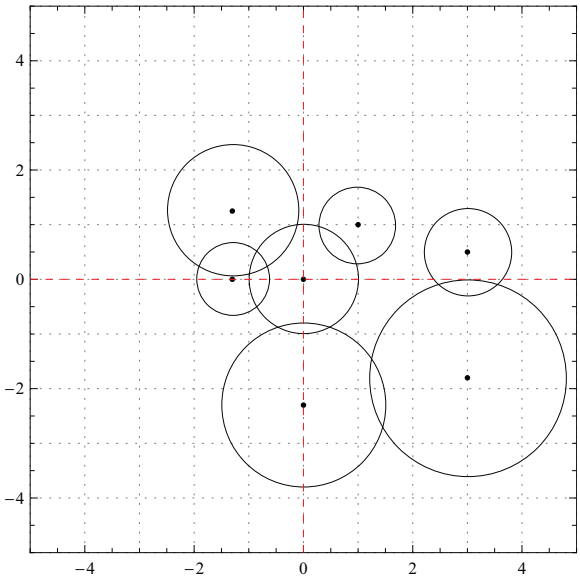
4.4 Algorithmes calculant le diagramme de VPUD

À la section 4.1, nous avons caractérisé algébriquement les régions de VPUD. Grâce à cela, nous avons mis en évidence les propriétés de telles régions. Dans la présente section, nous répondons à la question : comment calculer un diagramme de VPUD et le visualiser sur ordinateur ? L'exposé est structuré en trois sous-sections.

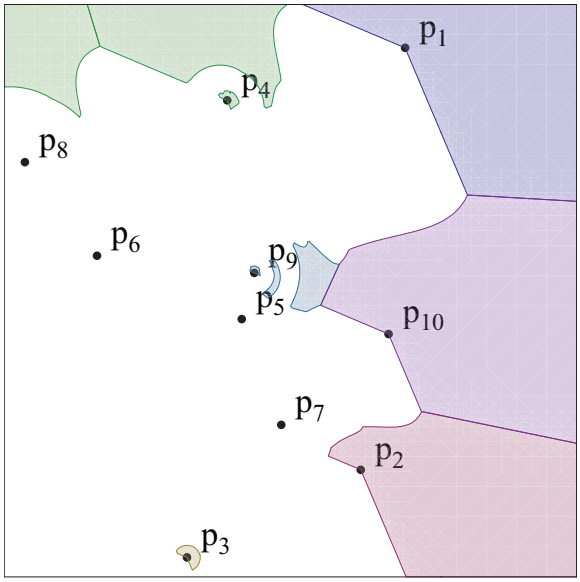
Pour commencer, nous expliquons en quoi *se baser sur la frontière des régions*, comme on a coutume de le faire, est problématique dans notre contexte. Nous donnons quelques exemples illustrant la difficulté à utiliser une telle représentation machine ici, et justifions ainsi l'adoption de représentations machines autres.

Dans la deuxième sous-section, la représentation adoptée pour encoder les régions est une *approximation discrète basée sur des points isolés*. Nous proposons deux algorithmes de recherche, opérant point par point, pour approximer un diagramme de VPUD complet. Ce dernier diagramme étant, à notre connaissance, un concept original *les algorithmes présentés sont les tout premiers algorithmes conçus à réaliser un tel calcul*.

Dans la troisième sous-section, la représentation adoptée pour encoder les régions est une *approximation intérieure basée sur des pavés*. La notion de pavé est issue de l'*arithmétique des intervalles* ; une théorie mathématique que l'on introduira brièvement. Nous proposons un algorithme opérant sur des pavés par bissection récursive pour calculer un diagramme de VPUD.



(a) *ensemble de disques utilisé pour calculer le diagramme de VPUD*



(b) *régions de VPUD associées aux germes de P, représentées en couleur*

FIGURE 4.8 – *Exemple de non-connexité des régions de VPUD. La région $V_B(p_9)$ a la particularité d'être non-connexe*

4.4.1 Difficultés inhérentes à la représentation sur machine

Se baser sur la frontière des régions est une méthode couramment choisie en informatique pour encoder des régions du plan. C'est notamment la frontière des régions qui est utilisée pour encoder le diagramme de Voronoï classique.

Or, notre étude concernant la frontière et les propriétés des régions de VPUD (résultats fournis aux sections 4.2 et 4.3, respectivement) semble indiquer qu'une telle représentation ne serait pas appropriée ici car elle génère trop de difficultés. Nous détaillons ces difficultés dans la liste ci-dessous. Pour chaque *item*, nous faisons référence à une propriété des régions de VPUD et décrivons le problème soulevé pour le calcul sur ordinateur.

1. Une manière simple et efficace de représenter des régions planes sur ordinateur, consiste à utiliser un polygone que l'on encode sous la forme de liste des coordonnées des sommets.
   Or, on sait d'après la propriété 4.2.3, que la frontière d'une région de VPUD est une courbe non-linéaire (sauf cas particuliers). Pour contourner cette difficulté, en principe il est possible d'échantillonner la courbe-frontière. On remarque que l'équation caractéristique dont on dispose est sous forme implicite, et non sous forme paramétrique. Dans ce contexte, obtenir un échantillon de points « convenablement répartis » pour modéliser la courbe n'est pas trivial.
2. Tout de même, supposons que les courbes-frontières aient été échantillonnées convenablement. La frontière d'une région de VPUD étant elle même une composition de morceaux de frontières de sur-régions élémentaires, il reste à « couper » correctement ces morceaux et à les lier les uns aux autres. Cette « liaison » ne peut être faite correctement sans connaître la position des *points de jonctions* (définition 4.2.1). Calculer la position de ces points de jonction exige des calculs relativement complexes. Nous avons réalisé ce calcul, en utilisant la propriété 4.2.5 ainsi que des outils mathématiques connus : théorème de Bezout, *méthode des résultants* (voir articles de Collins (1971), Allgower et al. (1992)) et *méthode de Laguerre* (*cf.* article de Goedecker (1994)) pour rechercher les racines de polynômes de degré important. La méthode fournit les résultats attendus, nous avons néanmoins jugé inutile de la détailler ici du fait de sa sophistication déraisonnable et de sa complexité de mise en œuvre.
3. Supposons que l'on sache composer la frontière complète des régions de VPUD, en calculant les points de jonction et en « reliant entre eux » les bons morceaux de courbe. Il resterait le problème de distinguer l'intérieur et l'extérieur de la région VPUD ainsi encodée. Une région de VPUD pouvant être non bornée, ce problème n'est pas trivial.
4. Finalement, La propriété 4.3.4 de non-connexité implique qu'une approche naïve type *suivi de contour* (avec calcul éventuel de tangentes/points d'inflexion) ne suffirait pas à elle seule à saisir toute l'information d'une région de VPUD. Comment pourrait-on être sûr, en effet de n'avoir oublié aucune composante connexe de la région de VPUD ?

Pour résumer, que peut-on tirer des observations faites ci-dessus ? Premièrement, la modélisation informatique d'une frontière de région de VPUD est loin d'être évidente. Ensuite, distinguer l'intérieur et l'extérieur de la région de VPUD, quand bien même la frontière serait modélisée efficacement, constituerait un autre problème non trivial. Enfin, puisque les régions de VPUD peuvent être non connexes, procéder par suivi de courbe n'est pas une approche pertinente.

Pour toutes ces raisons, nous avons orienté nos recherches vers des représentations machine qui approximent *l'intérieur* des régions. Deux représentations approximatives basées sur l'intérieur des régions sont exposées dans les sections suivantes. Pour chaque représentation, nous proposons un ou deux algorithmes de calcul. Ces algorithmes permettent de visualiser le diagramme de VPUD sur ordinateur.

4.4.2 Approximation discrète basée sur des points isolés

Algorithme de recherche par force brute

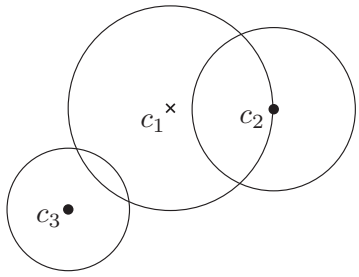
Dans cette sous-section nous présentons l'algorithme « naïf » pour calculer le diagramme de VPUD et le représenter sur machine. L'idée est la suivante : on se donne un jeu de données (nuage de points, ensemble de disques) et une boîte englobante. L'algorithme naïf discrétise la boîte englobante à l'aide d'une grille, où chaque intersection va correspondre à un pixel. On obtient alors une image approximative discrétisée des régions.

Rappelons quelles sont les entrées de l'algorithme :
- P un nuage de points constitué d'au moins deux points
- B un ensemble de disques standard
- z_1, z_2 deux points délimitant une boîte englobante
- la largeur de l'image en sortie (nombre de pixels)
- la hauteur de l'image en sortie (nombre de pixels)

```
P_ajuste ← ajuster P au repère de l'image
marquer de la couleur blanche tous les pixels de l'image
Pour x de 0 à largeur-1 faire
    Pour y de 0 à hauteur-1 faire
        Pour i de 0 à n-1 faire
            Si ( test_point((i, j), p_i, P_ajuste, B) == VRAI) Alors
                marquer le pixel (i,j) de la couleur associée à
                V_B(p_i)
                Retourner ;
            Fin Si
        Fin Pour
    Fin Pour
Fin Pour
```

Algorithme 1 – *Calcul par force brute du diagramme de VPUD (représentation point par point)*

L'algorithme 1 décrit le calcul en utilisant un langage pseudo-code.



(a) *EDS employé*

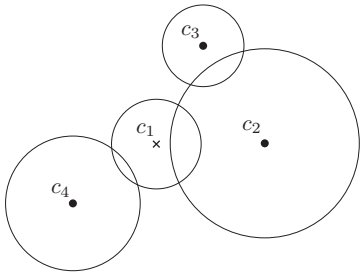
(b) *approximation du diagramme de VPUD*

FIGURE 4.9 – *Résultats produits par l'algorithme 1 force brute (jeu de données #1)*

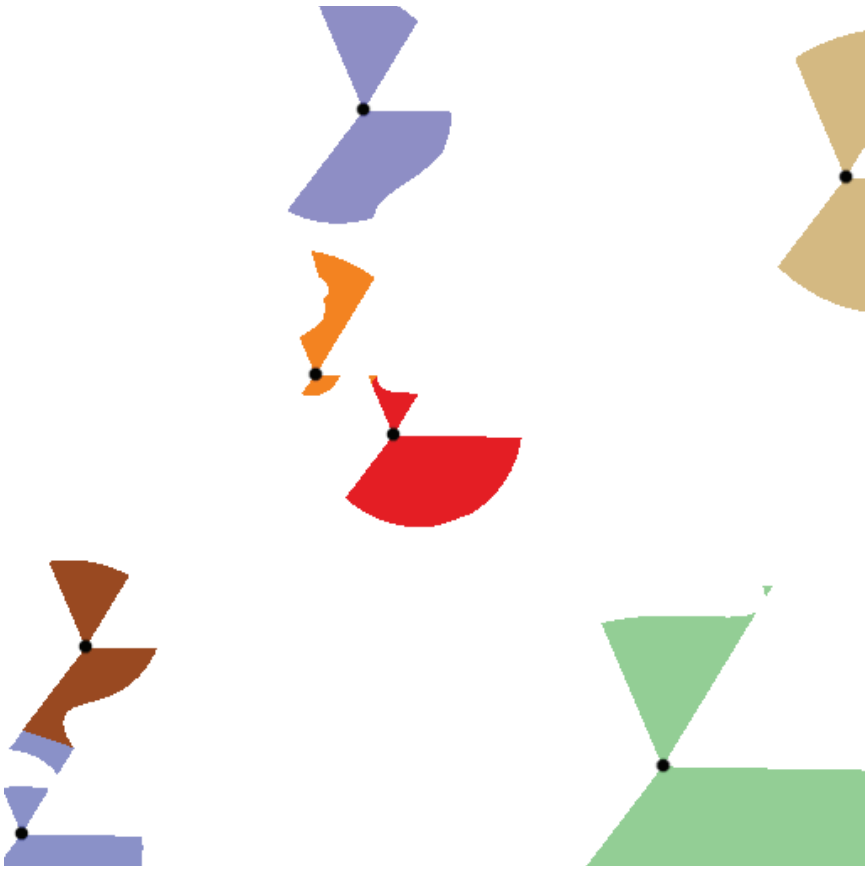
À l'aide de deux boucles imbriquées il procède pixel par pixel en testant à chaque fois si ce dernier appartient à l'une des régions de VPUD possibles. Ce test d'appartenance exige lui-même la vérification de toutes les inéquations du système algébrique équivalent, comme ce fut expliqué dans le théorème 4.1.6.

Nous complétons cette présentation en donnant quelques exemples de résultats obtenus sur ordinateur et un ordre de grandeur concernant leur coût en temps. On présente les résultats pour trois jeux de données distincts. Pour produire les images, nous avons choisi une résolution telle que $largeur = hauteur = 512$.

La figure 4.9 montre un diagramme obtenu pour un jeu de données comprenant 6 points et 3 disques. La machine de test qui a été utilisée est un PC portable équipé d'un processeur INTEL modèle $i3 - 350m$ cadencé

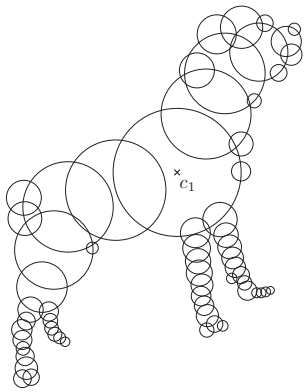


(a) *EDS employé*



(b) *approximation du diagramme de VPUD*

FIGURE 4.10 – *Résultats produits par l'algorithme 1 force brute (jeu de données #2)*



(a) *EDS employé*

(b) *approximation du diagramme de VPUD*

FIGURE 4.11 – *Résultats produits par l'algorithme 1 force brute (jeu de données #3)*

à $2GHz$. Le calcul de cette image a exigé le test de 262144 points, il fut réalisé en 29.8 secondes.

La figure 4.10 représente le diagramme obtenu pour le deuxième jeu de données constitué cette fois de 7 points et 4 disques. Comme la résolution était la même, l'algorithme a là encore testé 262144 points pour obtenir cette image. Le calcul a été réalisé en 30.9 secondes dans les mêmes conditions d'expérience.

Enfin, la figure 4.11 montre le diagramme obtenu pour un jeu de données comprenant 7 points et un EDS bien plus complexe cette fois-ci avec 61 disques. Une fois encore, l'algorithme a testé 262144 points pour dessiner l'image à la résolution 512×512. Le calcul a été réalisé en 319.6 secondes sur la même machine.

On voit que l'impact de la taille du jeu de données sur le temps de calcul est très fort, comme le prédit d'ailleurs la théorie (la complexité au pire est polynomiale). L'algorithme, bien que de plus en plus lent à mesure que les données sont volumineuses reste utilisable. Il a l'avantage d'être très simple à implémenter.

Cet algorithme 1 procède pixel par pixel et est par conséquent fortement ralenti lorsque l'on souhaite avoir une image de haute résolution. Si l'on considère le simple objectif de la visualisation des régions, le test d'un si grand nombre de pixels semble inutilement coûteux. Dans la partie qui suit nous utilisons un procédé de calcul probabiliste pour accéder à un affichage de qualité presque aussi bonne, en un temps nettement réduit.

Heuristique de visualisation rapide

On l'a vu tout à l'heure, l'algorithme de recherche par force brute teste un nombre de pixels très important. Or, tous les pixels n'ont pas besoin d'être testés si l'objectif est simplement de *visualiser* les régions de VPUD sur ordinateur.

Alternativement à la recherche systématique de pixels appartenant à la région de VPUD (stratégie coûteuse en temps), nous avons conçu un procédé de calcul non déterministe. Ledit procédé peut être qualifié d'*heuristique probabiliste*, en cela qu'il utilise des nombres pseudo-aléatoires durant son exécution et fournit une réponse imparfaite en un temps court.

Avant tout, récapitulons les entrées/sorties de cette heuristique 2. Les entrées sont les mêmes que pour l'algorithme force brute...

- P un nuage de points constitué d'au moins deux points
- B un ensemble de disques standard
- largeur de l'image en sortie (nombre de pixels)
- hauteur de l'image en sortie (nombre de pixels)

... avec deux paramètres additionnels qui sont spécifiques à l'heuristique :

- largeurZone, le nombre de pixels représentant la largeur d'une zone. Ce nombre doit être un diviseur de la largeur
- hauteurZone, le nombre de pixels représentant la hauteur d'une zone. Ce nombre doit être un diviseur de la hauteur

En plus de ces paramètres reçus en entrée, l'heuristique elle même est basée sur trois coefficients que l'on note α, β, γ. Nous allons revenir plus tard sur le sens de ceux-ci. Détaillons d'abord le principe de calcul, en donnant quelques-unes des clés de son fonctionnement.

Découpage en zones Pour calculer une approximation des régions de VPUD, l'image entière est d'abord découpée en plusieurs zones (8×8 zones dans nos batteries de tests). Chaque zone se voit attribuer un score, au commencement celui-ci est initialisé à 1.0.

Choix probabiliste de points à tester Contrairement à l'algorithme force brute vu précédemment, nous proposons ici de choisir les points à tester de manière probabiliste. On effectue donc un tirage pseudo-aléatoire en deux passes : choix de zone puis choix du pixel.

- Le premier tirage (implémenté par la fonction `choix_roulette`) est fait, non pas de manière équiprobable, mais de manière *pondérée* selon les scores respectifs des zones. Ainsi, une zone ayant un score relativement élevé a plus de chances d'être tirée au sort. À l'opposé, il devient impossible pour une zone de score nul d'être tirée au sort.
- Le deuxième tirage (fonction `choix_alea_pixel_zone`) est quant à lui purement aléatoire, il sélectionne un pixel au hasard dans une zone donnée.

Test et ajustement des scores Chaque pixel choisi est testé (test implémenté par la fonction), et si ce test révèle que le pixel appartient effectivement à une région de VPUD, le score de la zone correspondante est augmenté. On multiplie ledit score par le coefficient γ. Parallèlement à ce processus, le nombre de tests de chaque zone est compté. Lorsque ce dernier dépasse une certaine proportion (proportion fixée par le coefficient β), on considère que la zone a été suffisamment « examinée », elle ne doit plus être sélectionnée, c'est pourquoi le score 0 lui est affecté.

Terminaison précoce du calcul Deux conditions distinctes entraînent la fin du calcul. Soit le nombre de points testés est égal au nombre de pixels de l'image, auquel cas la boucle principale (boucle utilisant l'indice k) est complète et le programme se termine après un nombre d'itérations prévu à l'avance. En pratique, ce cas n'arrive jamais si les coefficients α, β, γ sont choisis de façon pertinente. Soit un nombre suffisant de zones a été testé de manière « intensive », ce qui a eu pour effet de faire chuter le score global (somme du tableau `scores`). Lorsque le score global décroît jusqu'à un certain seuil, déterminé par le coefficient α, nous mettons fin précocement au calcul. Nous venons de donner une description de quatre étapes clés de l'heuristique :

1. découpage en zones
2. sélection probabiliste de points à tester
3. test et ajustement des scores
4. terminaison précoce du calcul

Les coefficients α, β, γ jouent un rôle dans les étapes 3 et 4. Ils permettent de définir implicitement quel est le *compromis à réaliser* entre performance et qualité de l'image produite par l'heuristique. Par performance, on entend temps de calcul et plus précisément : nombre de tests à réaliser.

Sur la base de nombreuses expériences avec jeux de données variés, nous avons ajusté ces valeurs par essai/erreur pour déterminer de manière empirique, ce qui semble être un « bon compromis » :

$$\alpha = \frac{1}{2}$$
$$\beta = \frac{1}{3}$$
$$\gamma = 1,05$$

Le procédé de calcul complet est décrit en pseudo-code ci-dessous. Par la suite, nous utilisons le terme « heuristique de visualisation rapide » pour désigner ce dernier.

```
nbPixelsTotal ← (largeur × hauteur)
nbPixelsZone ← (largeurZone × hauteurZone)
nbZones ← (largeur/largeurZone) × (hauteur/hauteurZone)
déclarer un tableau de flottants nommé scores, de taille nbZones
déclarer un tableau d'entiers nommé nbTestsZone, de taille nbZones
Pour i de 0 à nbZones-1 faire
    scores[i] ← 1.0
    nbTestsZone[i] ← 0
Fin Pour
seuilScoreTotal ← α × somme(scores)
seuilNbTestsZone ← β × nbPixelsZone
Pour k de 0 à nbPixelsTotal-1 faire
    zoneChoisie ←choix_roulette(scores)
    Si ( nbTestsZone[zoneChoisie] > seuilNbTestsZone ) Alors
        scores[zoneChoisie] ← 0
        Si ( somme(scores) < seuilScoreTotal) Alors
            Terminer le calcul
        Fin Si
        passer à l'itération suivante de la boucle courante
    Fin Si
    incrémenter nbTestsZone[zoneChoisie]
    (x, y) ← choix_alea_pixel_zone(zoneChoisie)
    Pour i de 1 à n faire
        Si ( test_appartenance((x, y), p_i, P, B) = VRAI ) Alors
            marquer (x,y) de la couleur associée à V_B(p_i)
            scores[zoneChoisie] ← scores[zoneChoisie] × γ
            sortir de la boucle courante
        Fin Si
    Fin Pour
Fin Pour
```

Algorithme 2 – *Visualisation rapide du diagramme de VPUD (représentation point par point)*

Comme pour l'algorithme 1, nous présentons trois exemples de diagrammes de VPUD calculés via cette heuristique. La résolution des images est la même que précédemment : $largeur = hauteur = 512$. Pour les zones utilisées par l'heuristique nous avons choisi : $largeurZone = hauteurZone = 64$. Le nombre 64 est bien un diviseur de 512 (le pré-requis est satisfait), il en résulte un découpage en 8×8 zones.

La figure 4.12 montre un diagramme obtenu pour le jeu de données numéro 1 (le même que celui utilisé pour les tests de l'algorithme force brute). Il comprend 6 points et un ensemble de 3 disques. Le calcul de cette image a exigé le test de 79565 points. Sur la même machine de test que celle décrite précédemment, le calcul a duré 6.9 secondes.

Rappelons que sur le même jeu de données, l'algorithme force brute a nécessité 29.8 secondes de calcul et a testé 262144 points. Les images produites sont similaires. Celles obtenues par l'heuristique sont de qualité légèrement inférieure, pour un temps de calcul six fois plus court ! Le gain de performances par rapport à l'algorithme force brute est donc considérable.

La figure 4.13 montre le diagramme obtenu pour le jeu de données numéro 2 déjà utilisé. Il comprend 10 points et un ensemble de 5 disques. Pour obtenir cette image, l'heuristique a dû tester 84992 points. Le calcul complet a duré 7.6 secondes.

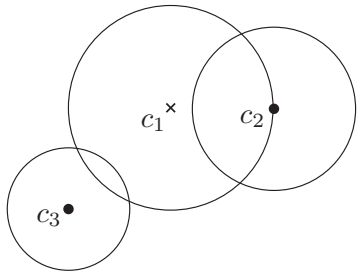
Rappelons que sur le même jeu de données, l'algorithme force brute a quant à lui nécessité 30.9 et a testé 262144 points, pour le même jeu de données. Le temps d'exécution de l'heuristique était environ quatre fois plus court sur cet exemple.

Enfin, la figure 4.14 représente le diagramme obtenu pour le jeu de données numéro 3 déjà utilisé. Ce dernier est constitué de 7 points et d'un ensemble de disques complexe composé de 61 disques. Pour obtenir cette image, l'heuristique a dû tester 85072 points et ce fut réalisé en 28.2 secondes.

On rappelle que l'algorithme force brute a nécessité le test de 262144 points, que son exécution a duré 319.6 secondes sur le même jeu de données. Sur cet exemple, le temps d'exécution de l'heuristique était environ onze fois plus court.

L'algorithme 1 de force brute, d'une part, et l'heuristique 2 de visualisation rapide d'autre part approximent tous deux la région de VPUD. Ils procèdent en testant des points isolés, ces point correspondant soit à l'ensemble des pixels d'une image (force brute) soit à des points aléatoirement choisis avec une pondération basée sur les scores et le découpage en zones (visualisation rapide). Ces deux procédés de calcul fonctionnent quelque soit le jeu de données choisi. Le calcul est relativement rapide, il s'exécute en temps raisonnable pour produire des images de 512×512 pixels. Ce temps est nettement réduit en utilisant l'heuristique plutôt que la recherche par force brute.

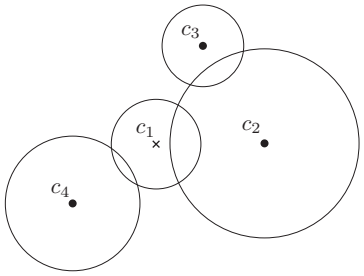
Avantages de l'approximation pixel par pixel. La représentation discrétisée (pixel par pixel) est satisfaisante pour les applications n'exigeant pas un calcul exact. Elle est très simple à implémenter et le temps de calcul peut être ajusté via la résolution que l'on choisit. Si l'affichage de détails



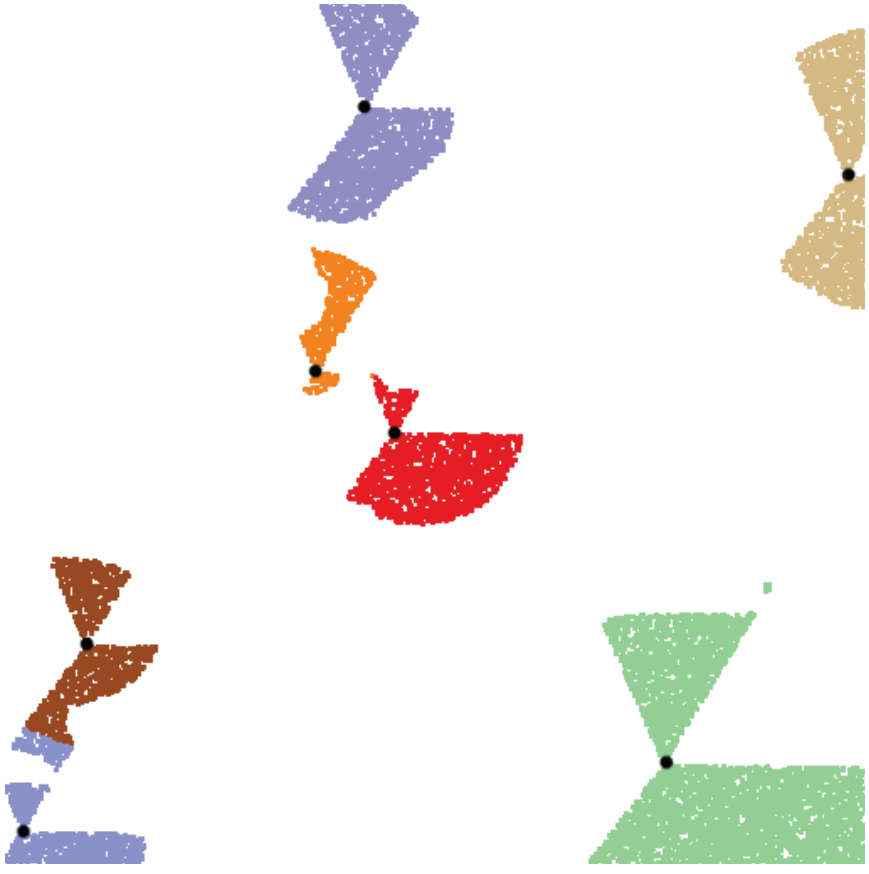
(a) *EDS employé*

(b) *approximation du diagramme de VPUD*

FIGURE 4.12 – *Résultats produits par l'heuristique 2 de visualisation rapide (jeu de données #1)*

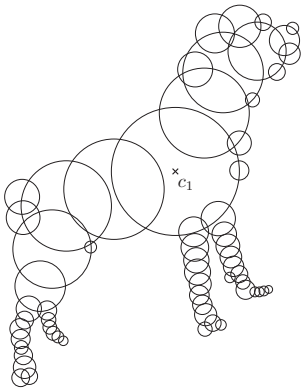


(a) *EDS employé*

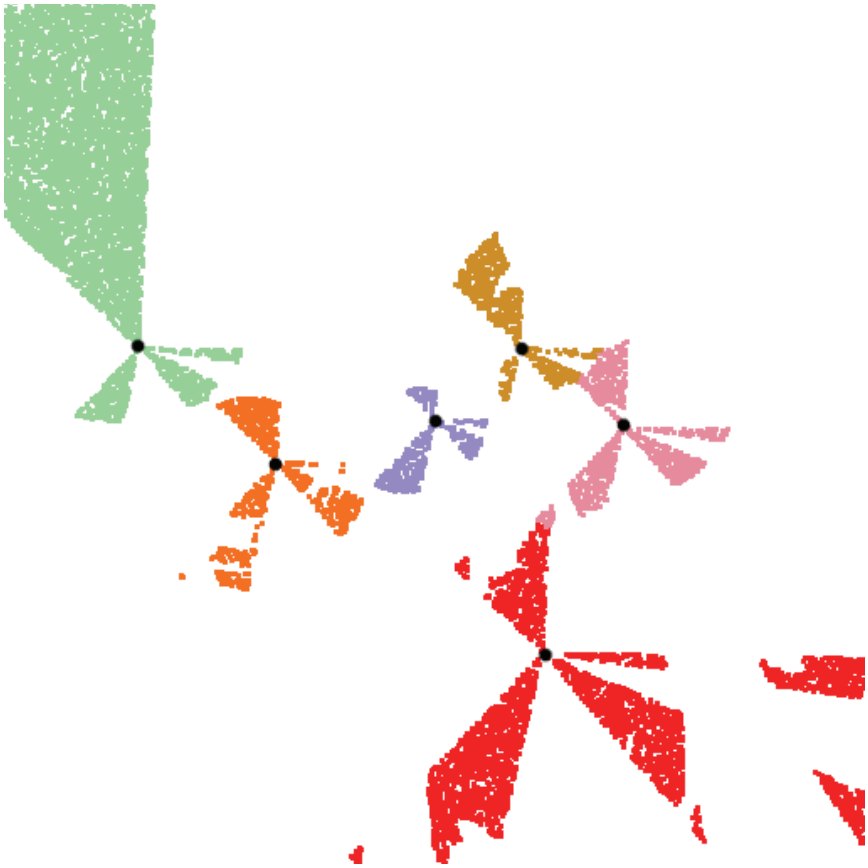


(b) *approximation du diagramme de VPUD*

FIGURE 4.13 – *Résultats produits par l'heuristique 2 de visualisation rapide (jeu de données #2)*



(a) *EDS employé*



(b) *approximation du diagramme de VPUD*

FIGURE 4.14 – *Résultats produits par l'heuristique 2 de visualisation rapide (jeu de données #3)*

fins est exigé, il suffit d'augmenter la résolution pour détecter les trous ou discontinuités.

Inconvénients de l'approximation pixel par pixel L'approximation proposée a peu de conséquences en pratique pour la seule *visualisation* des régions. En revanche elle peut être gênante pour des applications qui font appel aux propriétés des régions de VPUD. Prenons la planification de trajectoires comme exemple. L'ensemble des points testés étant discontinu, on n'a pas la garantie qu'entre deux points, le prédicat d'union de disques vide reste vérifié. Il n'est donc pas possible de déplacer de façon continue un point dans une région de VPUD ainsi calculée, en étant sûr que le prédicat d'union de disques vide reste vrai.

Nous avons exploré d'autres possibilités de représentation de régions de VPUD sur machine. On peut considérer non plus des points isolés dans le plan mais des *intervalles bidimensionnels* ou « pavés ». En appliquant les techniques *d'arithmétique des intervalles*, (une technique d'analyse numérique déjà bien étudiée) à notre problème de géométrie, nous avons obtenu un algorithme calculant les régions avec plus de précision. La sous-section suivante décrit les résultats obtenus suivant cette approche.

4.4.3 Approximation intérieure basée sur des pavés

Précédemment nous avons vu comment approximer les régions de VPUD en nous basant sur le test de points isolés. Ces points sont testés successivement, qu'ils aient été choisis un par un (algorithme force brute) ou de manière pseudo-aléatoire (heuristique de visualisation rapide).

Ci-dessous nous proposons une approximation d'un genre différent, approximation basée sur la notion de « pavé ». Dans un premier temps nous allons voir ce qu'est un pavé, et présenter l'*arithmétique des intervalles* ; domaine dont est issue cette notion. Ensuite nous allons décrire le procédé de calcul lui-même c'est-à-dire comment utiliser les pavés pour décrire les régions de VPUD.

Arithmétique des intervalles

L'arithmétique des intervalles désigne un ensemble de techniques d'analyse numérique employées par les scientifiques, depuis le travail fondateur de Moore (1962). Ce n'est que plus tard que l'arithmétique des intervalles prend vraiment son essor, grâce à l'augmentation de la puissance de calcul des ordinateurs, et à la mise en évidence des nombreuses applications possibles (voir les articles de Ratschek et Rokne (1988), Mitchell (1990), Jaulin et al. (2001), Michelucci et Foufou (2006) par exemple).

Ci-dessous nous rappelons quelques définitions utiles pour mettre en œuvre l'arithmétique des intervalles. Nous nous sommes inspiré de l'article didactique de Mitchell (1991) qui est une introduction au domaine, en même temps qu'un exemple d'application. En langue française, un cours synthétique a été proposé Révol (2004).

Intervalles et pavés : L'arithmétique des intervalles définit un système de calcul qui opère sur des éléments de base qui ne sont plus des nombres

« classiques » (entiers, rationnels, réels) mais des *intervalles*. Un intervalle représente toute une plage de valeurs. Un exemple est l'intervalle $[0\ ;\ 1]$ qui désigne tous les nombres réels compris entre 0 et 1. Plus généralement, tout intervalle est représenté par un couple de réels $[a, b]$, appelés *borne inférieure* et *borne supérieure* respectivement. La borne inférieure et supérieure peuvent être confondues, auquel cas l'intervalle dégénéré $[a, a]$ représente le réel a uniquement. Par extension, des intervalles multidimensionnels peuvent être utilisés. Dans notre contexte, nous allons utiliser des intervalles de dimension deux. On parlera alors de *pavés* pour les distinguer des intervalles de dimension un. Prenons un exemple : soient $I_x = [2.2\ ;\ 3]$ et $I_y = [-3.15\ ;\ 3.15]$ deux intervalles, leur produit cartésien donne le *pavé* $A = I_x \times I_y = [2.2\ ;\ 3] \times [-3.15\ ;\ 3.15]$.

Extension des opérations usuelles : L'arithmétique des intervalles étend les opérations arithmétiques usuelles (addition, soustraction, multiplication, division) aux intervalles. Considérons deux intervalles X,Y par exemple, il est possible de calculer la somme $(X + Y)$ qui est aussi un intervalle. La redéfinition des opérations appliquées aux intervalles est donnée ci-dessous. Dans le cas de pavés, le principe est le même pour chaque coordonnée.

Définition 4.4.1 (opération entre intervalles). Le résultat d'une opération $X <> Y$ réalisée entre deux intervalles est le plus petit intervalle ayant pour sous-ensemble l'ensemble :

$$\{x <> y \mid x \in X, y \in Y\}$$

Autrement dit, le résultat d'une opération entre deux intervalles X et Y est le plus petit intervalle contenant les résultats de l'opération appliquée à tous les éléments possibles de $x \in X$ et tous les éléments $y \in Y$ pris deux à deux.

En appliquant la définition précédente (abstraite mais générale) aux opérations arithmétiques usuelles, ces dernières se généralisent aux intervalles. On obtient les formules suivantes qui sont plus utiles en pratique.

$$\begin{aligned}
[a,b] + [c,d] &= [x+y, x+y] \\
[a,b] - [c,d] &= [a-d, b-c] \\
[a,b] \times [c,d] &= [min(ac,ad,bc,bd), max(ac,ad,bc,bd)] \\
[a,b]/[c,d] &= [a,b] \times [1/d, 1/c] \text{ à condition que } 0 \notin [c,d] \\
[a,b]^2 &= [min(a^2,b^2), max(a^2,b^2)] \text{ si } 0 \notin [a,b] \text{ et } [0, max(a^2,b^2)] \text{ sinon} \\
\sqrt{[a,b]} &= [\sqrt{a}, \sqrt{b}] \text{ si } 0 \leq a
\end{aligned}$$

Quelques cas particuliers peuvent être distingués :

- si a, b, c, d sont tous positifs alors $[a, b] \times [c, d] = [ac, bd]$
- la réciproque de la multiplication n'est pas définie si 0 appartient à $[c, d]$. En pratique, selon le domaine d'application, il est parfois plus intéressant de retourner un intervalle spécial représentant $]-\infty, +\infty[$ plutôt que de signaler une erreur

Remarque. Attention, on attire l'attention sur le fait que les propriétés algébriques de ces opérations ne sont pas les mêmes que celles des opérations associées aux nombres. En particulier : la soustraction n'est pas la réciproque de l'addition ; la division n'est pas la réciproque de la multiplication ; le résultat d'une multiplication d'un intervalle par lui même n'est pas égal à son élévation au carré ; la multiplication n'est pas distributive par rapport à l'addition.

Des opérations de calcul plus complexes, comme l'application de fonctions transcendantes, se généralisent aussi aux intervalles. Ainsi, trouver l'encadrement exact de n'importe quelle fonction monotone est trivial, par exemple :

$$e^{[a,b]} = [e^a, e^b]$$
$$[a,b]^3 = [a^3, b^3]$$

Évaluation d'expressions algébriques avec intervalles : Les opérations vues précédemment, qui s'appliquent sur des intervalles, peuvent être combinées entre elles pour évaluer une expression algébrique. Illustrons ce propos en évaluant, par exemple, l'expression polynomiale $10X^2 - 2X + 1$ pour l'intervalle $X = [0,2]$:

$$10X^2 - 2X + 1 = 10[0,2]^2 - 2[0,2] + 1$$

en appliquant les règles de calcul données précédemment, on obtient :

$$\begin{aligned} &= 10[0,4] - 2[0,2] + 1 \\ &= [0,40] - [0,4] + 1 \\ &= [-4,40] + 1 \\ &= [-3,39] \end{aligned}$$

Ainsi, on déduit que l'intervalle $[-3,39]$ contient *l'image* de l'intervalle $[0,2]$ par la fonction polynomiale $f(x) = 10x^2 - 2x + 1$. Une telle évaluation d'expression produit un résultat qui n'est pas toujours exact mais dont on sait qu'il contient l'image. On appelle « extension naturelle » une telle évaluation.

Extension naturelle de fonctions : Précisons la nature du résultat obtenu précédemment. *L'image d'un intervalle* Z par une fonction mathématique f est définie comme étant l'ensemble des résultats possibles de f appliquée à tous les éléments z de Z. Formellement, l'image $f(Z)$ est *égale* à l'ensemble : $\{f(z) \mid z \in Z\}$. Dans le cas général, déterminer cet ensemble est loin d'être évident. En revanche, un sur-ensemble de celui-ci peut être calculé en étendant les opérations de calcul aux intervalles.

Définition 4.4.2 (Extension naturelle des fonctions). Étant donnée une fonction rationnelle r, *l'extension naturelle* de r à un intervalle $X = [x_0, x_1]$ est définie comme l'évaluation de l'expression algébrique r pour l'intervalle X en utilisant les opérations d'arithmétique des intervalles définies précédemment. L'intervalle qui résulte de cette évaluation, noté $r[X]$

contient l'image exacte de la fonction réelle r appliquée sur tout l'intervalle X. Formellement :

$$r[X] \supset r(X) = \{r(x) \mid x \in X\}$$

Pour résumer : on vient d'introduire l'arithmétique des intervalles, de rappeler la notion d'intervalle et de pavé, les opérations applicables sur de tels objets. La possibilité de fournir un résultat contenant à coup sûr une valeur ou un ensemble de valeurs recherché fait que l'on peut parler de résultats garantis ou encore certifiés. Grâce à l'arithmétique des intervalles, il devient donc possible de calculer non plus avec des nombres uniquement, mais avec des ensembles.

Dans la partie suivante, nous proposons un algorithme d'approximation intérieure des régions de VPUD en nous servant de cette technique.

Algorithme opérant par bissection récursive

Au cours de nos travaux, nous avons utilisé la notion de pavé que l'on vient de voir, pour approximer l'intérieur de régions de VPUD et ainsi calculer région par région, un diagramme complet. Pour construire l'ensemble adéquat de pavés, nous avons utilisé la caractérisation algébrique des régions de VPUD (théorème 4.1.6), l'extension naturelle de fonctions (définition 4.4.2) et un procédé de bissection récursive des pavés.

Ci-dessous, nous détaillons ledit procédé et chaque étape de calcul. Nous énonçons ensuite l'algorithme complet en langage pseudo-code et donnons quelques exemples de résultats produits.

Avant toute chose, pour chaque point $p_i \in P$, il convient de déterminer le système d'inéquations décrivant la région $V_B(p_i)$. Ce système fournit un ensemble de fonctions caractéristiques $f_{i,j}(x, y)$ avec lesquelles le calcul devient possible. On sait qu'un point $M(x, y)$ appartient à la région de VPUD, si et seulement si toutes les inéquations de type $f_{i,j}(x, y) \geq 0$ sont satisfaites.

Le procédé de bissection récursive, au cœur de l'algorithme proposé est le suivant : on part d'une boîte englobante, qui est donc modélisable par un pavé. Notons ce pavé A. Les fonctions caractéristiques sont évaluées pour le pavé A tout entier, en appliquant les définitions d'arithmétique des intervalles vues plus tôt. Le résultat obtenu $f_{i,j}[A] = [b_{inf}, b_{sup}]$ est un intervalle. Trois cas de figures sont alors à envisager :

- Soit la borne supérieure b_{sup} est inférieure à 0, auquel cas il n'existe aucune valeur de l'intervalle $f_{i,j}[A]$ satisfaisant l'inéquation. Par conséquent, la région modélisée par le pavé A est *disjointe* de la région $V_B(p_i)$.
- Soit la borne inférieure b_{inf} est supérieure ou égale à 0, auquel cas toute valeur de l'intervalle $f_{i,j}[A]$ satisfait l'inéquation. Dans ce cas, on a l'implication suivante : Si les autres inéquations sont elles aussi satisfaites *alors* la région modélisée par le pavé A est *incluse* dans la région $V_B(p_i)$.
- Dans tout autre cas, il est impossible de conclure directement quant à l'existence ou la non-existence, au sein du pavé A, de points membres de la région $V_B(p_i)$.

Si l'on ne parvient pas à déterminer le caractère disjoint ou inclus d'un pavé (premier et deuxième cas de figure), nous « découpons » le pavé en quatre sous-pavés de taille identique et appliquons le même traitement sur chaque sous-pavé. Ce calcul récursif est appliqué jusqu'à atteindre une taille de pavé limite, fixée à l'avance.

Ci-dessous, nous énonçons en langage pseudo-code l'algorithme dont on vient d'expliquer le principe. Pour plus de lisibilité dans nos programmes, nous avons fait le choix de représenter les pavés par un point (indiquant le coin supérieur gauche du pavé) et une taille. L'algorithme lui-même repose sur trois fonctions auxiliaires. Nous les listons ci-dessous, de la plus simple à la plus complexe et en résumant leurs traitements respectifs :

- la fonction *colorier(xinf,yinf,taille,couleur)* sert à marquer le pavé dont le coin supérieur-gauche est positionné en (x, y), de largeur et hauteur égale à *taille*, comme étant inclus dans une région de VPUD. À chaque région de VPUD est donc associée une et une seule couleur.
- la fonction *fcar_vpud(i,j,k,xinf,yinf,taille)* implémente le calcul d'une fonction $f_{i,j}(x, y)$ comme décrite précédemment. Une telle fonction provient du système d'inéquations caractérisant la région $V_B(p_k)$.
- la fonction *test_appartenance(k,xinf,yinf,taille)* évalue l'appartenance d'un pavé à la région $V_B(p_k)$ en faisant appel au calcul de l'extension naturelle de chacune des fonctions caractéristiques $f_{i,j}$. C'est au cours de l'exécution de *test_appartenance* que l'on détermine dans lequel des trois cas de figure (cas de figure présentés ci-dessus) on se trouve en analysant le pavé décrit par les paramètres *xinf,yinf,taille*.

L'algorithme 3 présente ce procédé de calcul récursif basé sur les pavés, la fonction `testAppartenance` est elle aussi détaillée ci après.

```
Procédure analysePaveRec( xinf, yinf, taille, tailleMinimale : entier)
    Si (taille<tailleMinimale) Alors
        Retourner;
    Fin Si
    infoComplete = VRAI
    Pour k de 0 à n-1 faire
        resultat_test ← test_appartenance(k,xinf,yinf,taille)
        Si (resultat_test = INCONNU) Alors
            infoComplete ← FAUX provoquer la sortie de la
            boucle
        Fin Si
        Si (resultat_test = VRAI) Alors
            colorier(xinf,yinf,taille, tabCouleurs[k] ) Retourner;
        Fin Si
    Fin Pour
    Si (infoComplete = VRAI) Alors
        colorier(xinf,yinf,taille,BLANC) Retourner;
    Fin Si
    analysePaveRec(xinf, yinf, taille/2, tailleMinimale)
    analysePave(xinf+taille/2, yinf, taille/2, tailleMininimale)
    analysePave(xinf, yinf+taille/2, taille/2, tailleMininimale)
    analysePave(xinf+taille/2, yinf+taille/2, taille/2, tailleMinimale)
Fin
```

Algorithme 3 – *Approximation du diagramme de VPUD par bissection récursive et arithmétique des intervalles*

```
Fonction testAppartenance( k, xinf, yinf, taille : entier) : entier
    Pour i de 0 à m-1 faire
        Pour j de 0 à n-1 faire
            Si (i ≠ 0 OU j ≠ i) Alors
                interv_resultat ← fcar_vpud(i,j,k,xinf,yinf,taille)
                Si ( interv_resultat.borne_sup < 0) Alors
                    Retourner FAUX;
                Fin Si
                Si ( interv_resultat.borne_inf < 0) Alors
                    Retourner INCONNU;
                Fin Si
            Fin Si
        Fin Pour
    Fin Pour
    Retourner VRAI;
Fin
```

Algorithme 4 – *pseudo-code pour la fonction testant l'appartenance d'un pavé à la région de VPUD associée au point* p_k

Pour tester l'algorithme que l'on vient d'énoncer, et à des fins de pro-

totypage, nous l'avons d'abord testé des régions du plan parfaitement connues.

Par exemple, nous avons testé ce principe pour représenter le disque unité. Le résultat correspond bien à ce qui est attendu, comme illustré sur la figure 4.15a. Autre exemple : nous avons appliqué l'algorithme pour calculer un diagramme de VPUD classique. Le résultat produit est représenté figure 4.15b. Sur cet exemple là, ce sont donc trois régions de Voronoï différentes qui sont approximées simultanément via l'arithmétique des intervalles.

L'algorithme fonctionne indifféremment sur de telles régions simples, comme sur des régions de VPUD. La figure 4.16 illustre le résultat produit par l'algorithme pour calculer un diagramme de VPUD complet constitué de cinq régions.

En utilisant suffisamment de pavés, les solutions produites par l'algorithme 3 ont l'avantage de pouvoir modéliser les régions de VPUD quelque soit leur complexité (régions non connexes, avec trous, frontières de forme quelconque, *etc.*) et la précision souhaitée. L'algorithme proposé est donc parfaitement général en cela qu'il fonctionne sur tout jeu de données P, B.

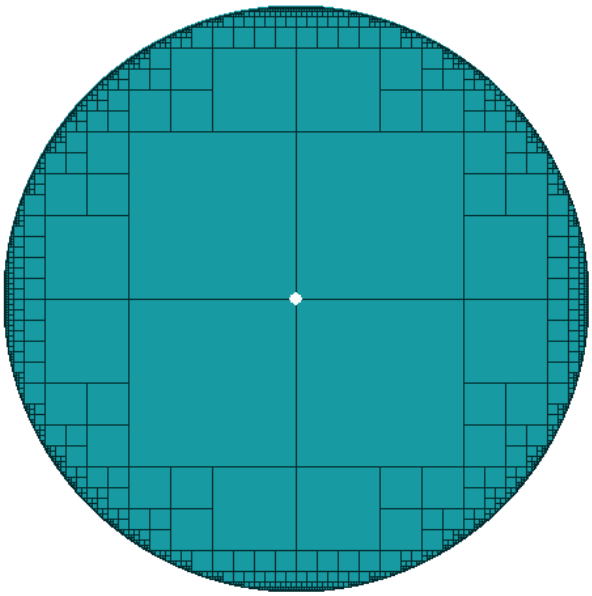
De plus, comme l'algorithme sélectionne uniquement les pavés dont on est sûr qu'ils sont membres de la région de VPUD, la solution produite est une approximation intérieure *garantie*. Tout point membre de l'approximation est nécessairement membre de la région de VPUD. C'est l'atout majeur de cette méthode, du fait que le calcul par intervalle donne un résultat garanti.

Finalement, le procédé de bissection récursive donne lieu à un ensemble de pavés qui peuvent être organisés selon une structure de données *quadtree*. Une telle structure est particulièrement efficace en termes de temps de calcul, son utilisation est déjà courante pour optimiser les algorithmes associées à diverses applications notamment la synthèse d'images.

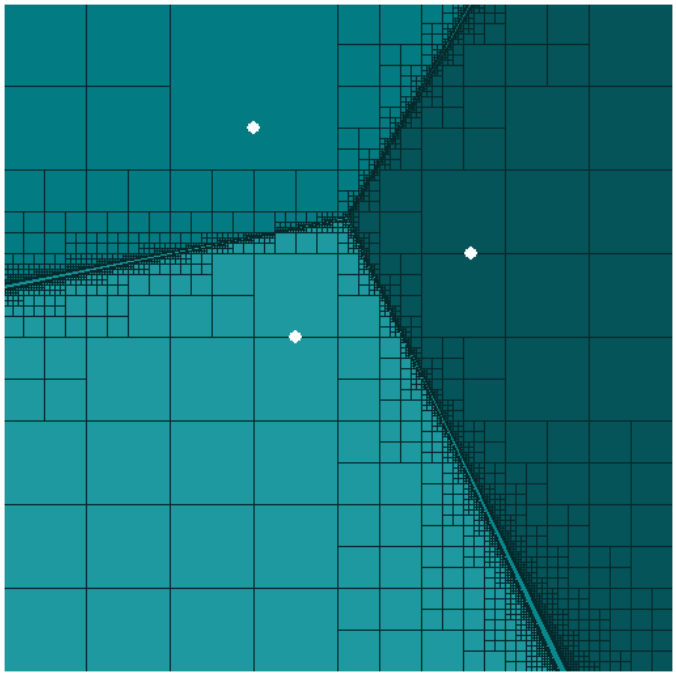
Conclusion du chapitre

Dans un premier temps, nous avons montré que les régions de Voronoï Paramétrées par une Union de Disques (abrégé régions de VPUD) sont égales à l'intersection de multiples sur-régions élémentaires. Ensuite, on a vu qu'une sur-région élémentaire peut se définir via une unique inéquation cartésienne. Nous basant sur ces deux faits, nous avons caractérisé algébriquement les régions de VPUD. Nous avons ainsi obtenu une définition des régions de VPUD plus utile en pratique.

Grâce à cette redéfinition des régions de VPUD, nous avons étudié les régions de VPUD et les sur-régions constituantes pour mettre en évidence plusieurs propriétés mathématiques globales. Ces propriétés (existence de points de jonction dans la frontière d'une région de VPUD, non-linéarité de la frontière de sur-régions élémentaires, non-connexité des régions de VPUD dans le cas général, *etc.*) sont à prendre en compte pour concevoir des algorithmes de calcul et de visualisation des régions de VPUD. Pour notre part, nous avons proposé trois algorithmes permettant de visualiser sur machine et en temps raisonnable n'importe quel ensemble de régions de VPUD.

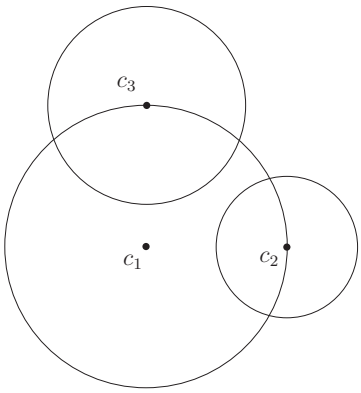


(a) *approximation d'un disque*



(b) *approximation de régions Voronoï*

FIGURE 4.15 – *Résultats de l'algorithme 3 appliqué à des régions bien connues, décrites par une ou plusieurs inéquations algébriques*



(a) *EDS employé*

(b) *approximation du diagramme de VPUD*

FIGURE 4.16 – *Résultat produit par l'algorithme 3 : approximation de diagramme de VPUD par bissection récursive de pavés. Pour ne pas gêner la visualisation, les lignes du découpage en pavés ne sont pas représentées ici*

Les deux premiers algorithmes (recherche par force brute, heuristique de visualisation rapide) opèrent point par point. Ils se révèlent efficaces pour la simple tâche de visualisation mais fournissent peu de garanties mathématiques quant à la validité topologique des résultats.

Le troisième algorithme (bissection récursive) opère par juxtaposition de « pavés » de différentes tailles. La représentation utilisée est toujours discrète globalement, mais est continue par morceaux. Cet algorithme est plus coûteux mais il est le seul qui, grâce aux outils de l'arithmétique des intervalles, offre des garanties mathématiques sur topologie des régions calculées. De telles garanties sont indispensables pour des applications du diagramme de VPUD autres que la simple visualisation.

Voilà qui clôt la partie consacrée à notre généralisation du diagramme de Voronoï. La deuxième partie du mémoire traite d'un thème *a priori* différent : le placement de formes géométriques. En réalité, la connexion forte entre ces deux thèmes nous a permis de proposer plus récemment une piste pour le *calcul géométrique exact* des régions de VPUD. Cette piste de recherche est exposée en annexe A.7.

Deuxième partie

Placement de formes géométriques en présence d'obstacles

Placement d'union de disques dans un nuage de points

5

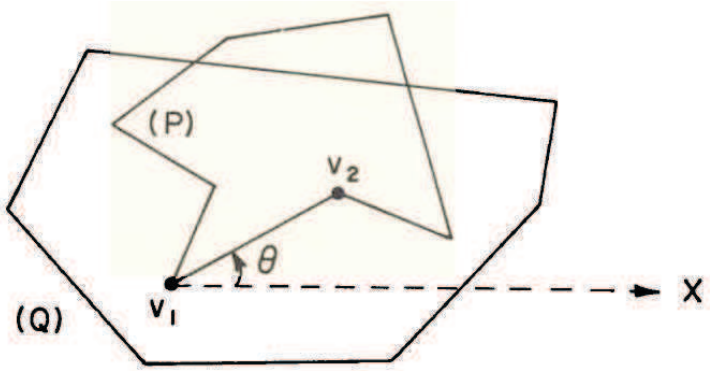
Sommaire

5.1 Travaux existants en placement de formes géométriques 93
5.1.1 Motivations pour l'étude du placement de formes géométriques . . . 94
5.1.2 Comparaison des problèmes de placement . . . 97
5.1.3 Placement et packing . . . 102
5.2 Placement d'union de disques : définition et notations 103
5.3 Espace de configurations . . . 104
5.4 Problèmes de placement traités dans la thèse . . . 106
5.4.1 Placement par rotation uniquement . . . 106
5.4.2 Placement par homothétie uniquement . . . 107
5.4.3 Placement par homothétie et rotation combinées . . . 109
5.4.4 Placement par translation . . . 112
5.4.5 Cas particulier d'un espace de configuration fini . . . 112
Conclusion du chapitre . . . 113

Où l'on donne un aperçu des travaux de recherche existants dans la littérature qui concernent le placement de formes géométriques dans un environnement avec obstacles. Nous comparons entre eux les problèmes de placement répertoriés selon quatre caractéristiques : représentation des obstacles, forme géométrique à placer, transformations considérées et objectif du placement.

Nous définissons ensuite une variante d'un tel problème de placement, variante où il s'agit de placer une union de disques dans un nuage de points en utilisant toute combinaison de translation, homothétie et rotation de sorte à réaliser un objectif de vacuité. Nous proposons un cadre global apte à décrire tout problème de ce type.

Grâce à ce cadre global, une combinaison de translation, homothétie et rotation particulière peut être spécifiée via un paramètre appelé *espace de configurations*. Plusieurs problèmes de placement d'union de disques sont ainsi unifiés et énoncés suivant le même formalisme. Nous le montrons sur quatre exemples : placement par rotation, par homothétie, par homothétie-rotation combinées et par translation.



Étant donnés deux polygones simples P et Q arbitrairement choisis, il s'agit de déterminer si Q peut contenir P et si oui, de fournir un placement pour P tel que Q contienne P. Dans ce contexte, un placement correspond à la donnée d'un couple : point v_1, angle $\theta = \angle\left(\overrightarrow{v_1X}, \overrightarrow{v_1v_2}\right)$

FIGURE 5.1 – « *Polygon containment problem* » *introduit par Chazelle*

5.1 TRAVAUX EXISTANTS EN PLACEMENT DE FORMES GÉOMÉTRIQUES

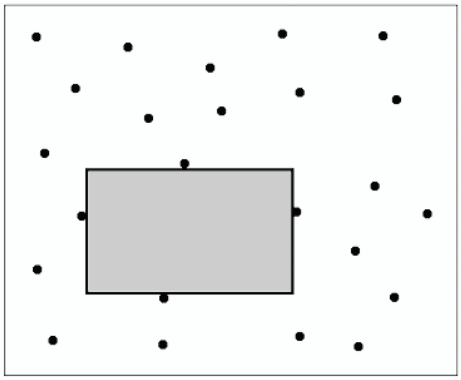
Le terme « placement de formes géométriques » désigne une famille de problèmes étudiés de manière récurrente en géométrie algorithmique. De manière très générale, la problématique est un « aménagement de l'espace » automatisé, grâce à des algorithmes de recherche, comme expliqué dans la présentation grand public de Parizot et Boissonnat (2004) disponible sur le *Web*.

Ci-dessous nous commençons par donner deux exemples de tels problèmes.

Exemple 1. L'article de Chazelle (1983) a été le premier à introduire le problème de placement appelé *Polygon Containment Problem*. Il s'agit, étant donné deux polygones simples arbitraires P et Q, de déterminer si Q peut contenir P en son intérieur et si oui, de fournir une configuration de P relativement à Q qui puisse réaliser cet objectif. On cherche donc à trouver au moins un placement d'un polygone dans un autre polygone, en utilisant les opérations de rotation et translation uniquement. Ce premier exemple est illustré sur la figure 5.1.

Exemple 2. Par la suite, les travaux de Chazelle et al. (1986) ont porté sur le problème dit du Maximum Empty Rectangle, ou « *MER problem* ». Ce dernier est illustré par la figure 5.2. Étant donné un rectangle englobant un nuage de points, il consiste à trouver dans ce rectangle englobant un autre rectangle d'aire maximale qui ne contienne lui-même aucun point.

Dans les sous-sections qui suivent, nous allons d'abord voir quelles sont les motivations pour l'étude de tels problèmes, avant de comparer ces problèmes les uns aux autres pour identifier leurs caractéristiques fondamentales.



Étant donné un rectangle R englobant un nuage de points, il s'agit de calculer le plus grand rectangle (rectangle d'aire maximale) inclus dans R, ayant ses côtés parallèles à ceux de R et ne contenant aucun point en son intérieur

FIGURE 5.2 – « *Maximal Empty Rectangle problem* » *étudié par Chazelle et al.*

5.1.1 Motivations pour l'étude du placement de formes géométriques

Plusieurs domaines concrets sont concernés par le *placement de formes géométriques*. Dans la littérature, nous avons relevé les mentions faites quant aux possibilités d'application dans l'industrie textile et métallurgique, la robotique et l'algorithmique, entre autres.

Dans cette sous-section, nous exposons quelques uns de ces exemples d'application concrète, afin de répondre à la question du : pourquoi étudier le placement de formes géométriques ?

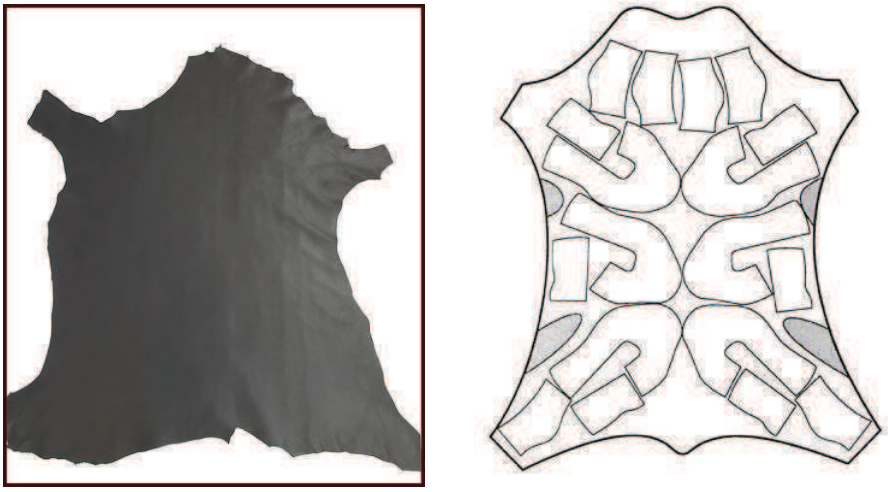
Applications dans le domaine de l'industrie

Les professionnels de l'industrie textile et/ou métallurgique sont parfois confrontés au problème suivant : partant d'une surface de matière première comportant des défauts et d'un motif à découper, quelle est la meilleure façon de découper une forme similaire au motif ne contenant aucun défaut ? Plus précisément, en maroquinerie où le motif est appelé gabarit, on découpe des morceaux de cuir de même forme que le gabarit, de sorte que la matière découpée puisse être cousue pour assembler l'objet à produire. Des défauts tels que des variations d'aspect ou des imperfections dues à la finition du cuir, gênent la découpe. Ce problème est illustré sur la figure 5.3.

Dans l'industrie métallurgique, comme expliqué dans le travail de Chew et Kedem (1989), on souhaite découper un carré de métal plein (sans trou) le plus grand possible, à partir d'une plaque contenant quelques petits trous. Il s'agit de trouver le « meilleur » emplacement pour le carré, en tenant compte des obstacles qui empêchent une découpe parfaite c'est-à-dire utilisant toute la surface de la plaque.

Applications en robotique

En robotique, les chercheurs étudient différentes stratégies pour résoudre le problème dit de la *planification de trajectoires*, comme expliqué



(a) *les matières premières possèdent souvent des défauts localisés qui sont gênants pour la fabrication*

(b) *lors de la découpe, on cherche à gaspiller le moins de matière possible tout en évitant les défauts*

FIGURE 5.3 – *Application du placement de formes géométriques dans l'industrie textile (maroquinerie)*

dans les travaux de Agarwal et al. (1997). Cette problématique est illustrée dans la figure 5.4.

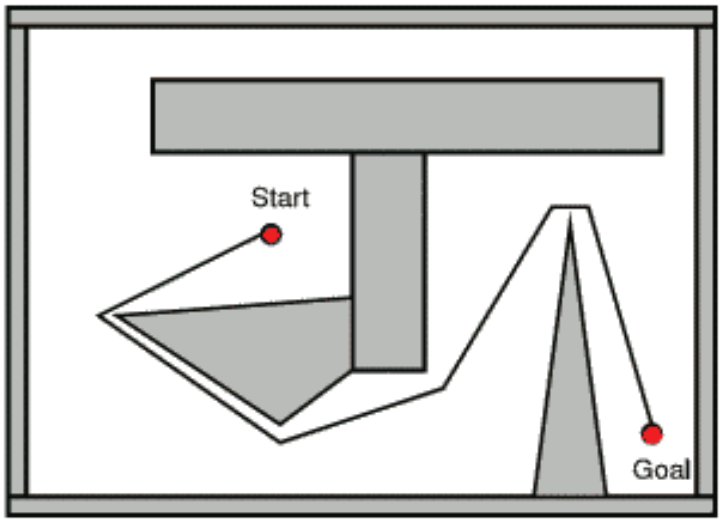
En résolvant un problème de placement, on peut chercher au moins un placement autorisé, mais il est parfois plus intéressant de connaître *tous* les placements autorisés. En calculant non pas une mais toutes les solutions de placement existantes, il devient possible par la suite de vérifier qu'une forme se déplaçant selon une trajectoire donnée n'entre pas en collision avec les obstacles. Ce problème a été étudié par le passé, notamment en employant deux disques, voir l'article de Hirsch et Halperin (2002).

Applications en algorithmique

Par analogie avec la planification de trajectoires, les simulations physiques sur ordinateur font appel à des algorithmes particuliers dits de *détection de collision*. Une présentation complète du sujet est faite dans le livre de Ericson (2005a). Nous illustrons le problème en 2D dans la figure 5.5.

À condition que les obstacles soient statiques, un calcul *a priori* de toutes les solutions de placement rend ici possible l'accélération des algorithmes. En effet, ces algorithmes peuvent alors se servir de résultats pré-calculés, et réaliser ainsi des économies en temps de calcul lors de tests d'intersection entre objets géométriques. Un tel procédé améliore les performances de simulations temps réel comme celles produites dans le domaine du jeu vidéo.

Ainsi, on voit que pour le placement de formes géométriques, il est intéressant de grouper l'ensemble des solutions pour en faire une structure de données. Une telle structure, qui est un résultat pré-calculé, fournit la connaissance de l'ensemble de tous les emplacements libres. Elle encode dans une structure les tests de proximité nécessaires. Grâce à cela, on peut déterminer si un emplacement pour la forme géométrique est libre ou non *via* un simple test d'appartenance. Une fois la structure de données



La planification de trajectoires ou motion planning *consiste, étant donnés un point de départ et une destination, à déterminer pour un mobile ponctuel un déplacement continu entre ces points, de manière à éviter les obstacles statiques. Une version plus difficile du problème considère un mobile non ponctuel, doté d'une forme géométrique connue éventuellement complexe*

FIGURE 5.4 – *Application du placement de formes géométriques en robotique : planification de trajectoires*

construite, un nouveau calcul est possible, celui-ci équivaut à « prédire » qu'une forme placée dans telle ou telle configuration ne contient obstacle. De tels algorithmes économisent leur temps de calcul en réutilisant l'information pré-calculée.

On vient de voir que les motivations pour étudier le placement de formes géométriques sont multiples et diverses. La section qui suit propose un aperçu des travaux existants et compare les problèmes répertoriés entre eux.

(a) *Pas de collision* (b) *Collision détectée*

FIGURE 5.5 – *Détection de collision en 2D entre formes géométriques simples*

référence	obstacles	forme géom.	opérations
Chazelle (1983)	poly. convexe	poly. convexe	T.R.
Toussaint (1983)	nuage de points	cercle	T.H.
Naamad et al. (1984)	nuage de points	rectangle	T.H.
Fortune (1985)	nuage de points	cercle	T.H.
Chew et Drysdale (1986)	nuage de points	cercle	T.H.
Chazelle et al. (1986)	rectangle et points	rectangle	T.H.
Leven et Sharir (1987b)	poly.s simples	poly. convexe	T.
Sharir (1987)	polyhèdre	polyhèdre convexe	T.
Avnaim et Boissonnat (1988)	poly.s	poly. simple	T.R.
Chew et Kedem (1989)	poly.s simples	poly. convexe	T.H.R.
Sharir et Toledo (1994)	poly.s simples	poly. convexe	T.H.R.
Agarwal et al. (1996)	poly. convexe	poly. convexe	T.H.R.
Halperin et Sharir (1996)	poly. simple	poly. simple	T.R.
Agarwal et al. (1997)	poly.s	poly. convexe	T.R.
Díaz-Báñez et al. (2003)	nuage de points	couronne	T.H.

TABLE 5.1 – *Spécificités des problèmes de placement étudiés lors de travaux antérieurs.*

5.1.2 Comparaison des problèmes de placement

La sous-section précédente a présenté les motivations quant à l'étude de problèmes qui consistent à placer une forme géométrique au sein d'un espace contenant des obstacles. L'étude de tels problèmes a déjà donné lieu à un grand nombre de publications par la communauté scientifique. L'article de Chew et Kedem (1989) souligne le fait que les problèmes de placement de formes géométriques traités sont similaires les uns aux autres, sans pour autant être identiques.

Dans cette section, en nous basant sur cette observation, nous proposons une analyse des travaux existants qui porte sur quatre caractéristiques fondamentales. Celles-ci constituent autant de « points de vue » différents sur le problème, qui se focalisent sur :

- la forme géométrique de l'objet à placer ;
- le modèle des obstacles ;
- les transformations considérées pour le placement ;
- le ou les objectif(s) dépendant(s) des obstacles que l'on cherche à atteindre.

Ci-dessous, nous adoptons successivement un de ces quatre points de vue pour relever les différences entre problèmes traités par les travaux existants. Suivant le point de vue choisi, on peut tirer diverses observations de la comparaison de deux articles. C'est pourquoi l'on s'autorisera à citer à plusieurs reprises un même article.

La liste sans doublons des travaux auxquels nous faisons référence est récapitulée dans le tableau 5.1. Les lettres R., H., T. utilisées dans ce tableau sont des abréviations pour désigner les transformations géométriques : *rotation, homothétie et translation* respectivement. Pour des raisons de formattage, le mot « polygone » est aussi abrégé en *poly.* (ou *poly.s* au pluriel).

Représentation des obstacles

En comparant les travaux existants entre eux, on remarque que l'environnement dans lequel on réalise le placement n'est pas forcément de même nature. À chaque fois, un ensemble d'entités géométriques représente les obstacles, ces entités pouvant être très différentes d'un problème à l'autre.

Dans le cas du « polygon containment problem » introduit par Chazelle (1983), dont on a déjà parlé en début de section, l'étude porte sur la question de savoir si un polygone peut être placé à l'intérieur d'un autre. Dans ce cas l'entité qui fait obstacle est elle-même un polygone (unique).

Par contraste, pour le « MER problem » du même auteur, les entités faisant obstacle sont cette fois-ci un ensemble fini de points (ou nuage de points) mais aussi le rectangle « englobant ».

L'article de Avnaim et Boissonnat (1988) décrit quant à lui un problème intitulé « polygon placement under translation and rotation » qui représente les obstacles par un ensemble de polygones simples quelconques. La même représentation est adoptée pour le problème traité dans l'article de Chew et Kedem (1989).

Le tableau récapitulatif 5.1 recense plusieurs autres exemples. Ces différences entre problèmes traités dans la littérature, concernant la représentation des obstacles, sont illustrées par la figure 5.6.

Une fois que la représentation des obstacles est fixée, une définition de la forme géométrique à placer est requise pour pouvoir formaliser et traiter un problème de placement.

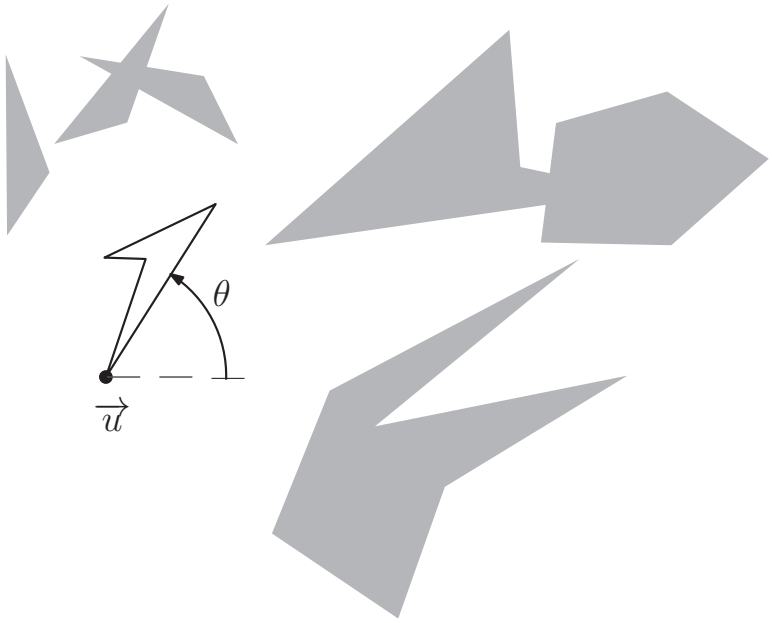
Forme géométrique à placer

Dans les travaux existants, la forme géométrique que l'on place peut-être plus ou moins complexe. Pour que le problème étudié ait une utilité en pratique, il est nécessaire que cette forme puisse modéliser un objet réel avec assez de précision.

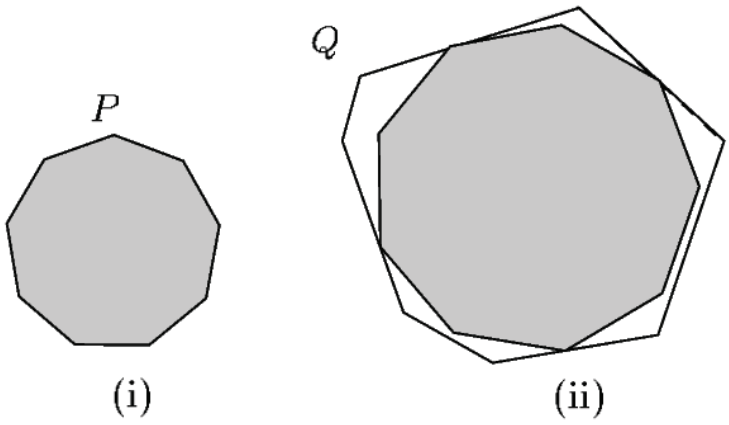
Plusieurs articles de placement dont ceux de Sharir et Toledo (1994), Agarwal et al. (1997) étudient le problème de placement d'une forme géométrique représentée par un polygone simple et convexe. Les algorithmiques proposés sont efficaces, néanmoins cette efficacité vient justement de la simplicité de la forme employée. Dans plusieurs applications, la représentation de la forme par polygone simple et convexe est insuffisante. Un problème concret manipulant un objet réel tel qu'un écrou (objet avec un trou), par exemple, ne pourrait être résolu algorithmiquement en reprenant la solution des articles mentionnés. De plus, les solutions proposées ne sont pas extensibles à des formes plus complexes, ou lorsqu'elles le sont c'est au prix d'une perte de simplicité de la méthode *cf.* l'article de Avnaim et Boissonnat (1988) décrivant le placement de polygones non convexes.

Pour d'autres applications spécifiques, il existe des travaux employant une forme géométrique différente telle que : le cercle (Toussaint (1983), Chew et Drysdale (1986), Augustine et al. (2010)) ; l'ellipse (Dwyer et Eddy (1994)) ; la couronne (Díaz-Báñez et al. (2003), voir la figure 5.7).

Ainsi, les travaux existants se distinguent par la forme géométrique à placer. Cette forme est aussi placée suivant une transformation qui peut

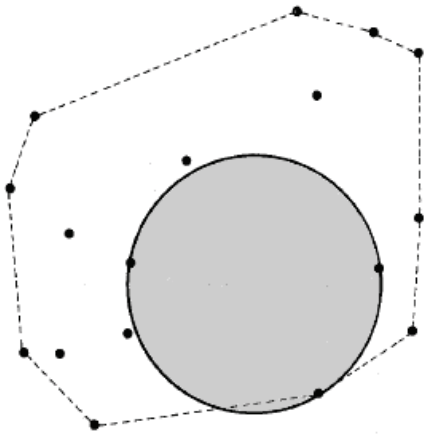


(a) *Placement d'un polygone dans un environnement polygonal ; problème étudié par Avnaim et Boissonnat (1988). Les obstacles sont un ensemble de polygones quelconques*

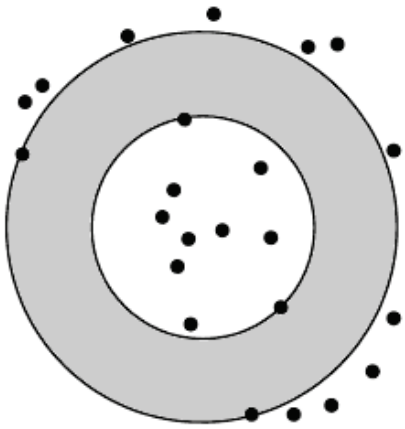


(b) *Placement d'un polygone convexe P (i) dans un autre (ii). Les obstacles sont les bords du polygone Q, problème étudié par Agarwal et al. en 1998*

FIGURE 5.6 – *Différence de représentation des obstacles, entre problèmes décrits dans la littérature*



(a) *placement d'un disque, étudié par Toussaint en 1983*



(b) *placement d'une couronne, étudié par Diaz-banez et al. en 2003*

FIGURE 5.7 – *Différence de forme géométrique à placer, entre problèmes décrits dans la littérature*

être toute combinaison de translation, homothétie et rotation. Nous détaillons cet aspect dans la partie suivante.

Transformations considérées

Les travaux existants sur le placement se distinguent également par la transformation utilisée pour placer la forme.

L'article de Fortune (1985) résout le *polygon containment problem* en utilisant uniquement la translation. L'article de Chazelle (1983) traite quasiment le même problème, cette fois-ci pour une combinaison de translation et rotation. L'algorithme proposé dans ce dernier article est, comme l'on peut s'y attendre, moins efficace que celui de Fortune. Cela s'explique par le fait que la transformation utilisée est plus puissante.

Pour obtenir des algorithmes efficaces, plusieurs travaux en rapport avec la planification de trajectoires (Sharir (1987), Leven et Sharir (1987a)) se restreignent à la translation seule. Tout de même, les travaux décrits par Avnaim et Boissonnat (1988), Halperin et Sharir (1996) considèrent le placement par translation et rotation combinées.

Le problème de placement dans sa généralité a été abordé dans l'article de Agarwal et al. (1998), il recherche un placement en utilisant toute combinaison de translation, homothétie, rotation.

Le choix des transformations considérées est donc délicat car il détermine un compromis fait entre deux avantages mutuellement exclusifs.

Le premier avantage est la simplicité et la performance algorithmique. Plus le nombre de transformations est réduit, plus l'algorithme est simple à concevoir et à mettre en œuvre.

Le second avantage est la capacité à donner lieu à de nombreuses applications concrètes. Plus le nombre de transformations est grand, plus le nombre de « questions pratiques » pouvant être traitées par l'algorithme est important. Ces questions pratiques exigent souvent l'utilisation d'une transformation complexe, combinant plusieurs transformations élémentaires.

On vient de voir qu'un problème de placement dépend des obstacles (polygones, nuage de points, *etc.*), de la forme à placer (polygone, cercle, couronne, ellipse) et de la transformation appliquée à celle-ci. Dans la partie suivante, nous reportons notre attention sur le but du placement lui-même qui est, une fois de plus, spécifique au problème.

Objectif du placement

Nous venons de voir plusieurs caractéristiques concernant les problèmes de placement que l'on trouve dans la littérature. Il s'avère que l'objectif même du placement est spécifique, comme expliqué ci-dessous.

Premièrement, pour certains travaux existants, il s'agit simplement d'éviter les obstacles. Cet objectif est le plus fréquent, il est celui qui permet par exemple de calculer où peut aller un robot sans heurter les murs. Dans sa formulation mathématique, éviter les obstacles revient à placer la forme géométrique de manière à ce que l'intersection ensembliste entre les obstacles et entre cette dernière soit égale à l'ensemble vide. Les articles de Avnaim et Boissonnat (1988), Agarwal et al. (1997) parmi beaucoup d'autres emploient cet objectif.

Deuxièmement, à la condition d'avoir une forme vide, de nombreuses études ajoutent la maximisation de la taille (*i.e.* maximiser la surface recouverte par la forme placée). Ceci peut être utile dans l'industrie et à nouveau en robotique. Les auteurs introduisant cet objectif pour la robotique expliquent notamment qu'une planification de trajectoire peut être correcte en théorie, mais que si le résultat est un chemin longeant continuellement les obstacles, elle sera peu utile en pratique (dû aux erreurs et à la précision imparfaite des senseurs installés sur les robots). Le terme anglais *extremal polygon containment problems* est employé dans la littérature pour désigner toute cette famille de problèmes. Une présentation synthétique en est faite par Sharir et Toledo (1994), Agarwal et Sharir (1998). Comme exemples indépendants, on peut citer : les articles décrivant le *Largest Empty Circle (LEC) problem cf.* Toussaint (1983), Chew et Drysdale (1986), le *Largest empty rectangle problem* introduit par Chazelle et al. (1986), le *Largest empty annulus problem* introduit par Díaz-Báñez et al. (2003), le problème de trouver le plus grand polygone convexe vide comme expliqué dans l'article de Chew et Kedem (1989), *etc.*

Dans cette sous-section, une comparaison des problèmes de placement de formes géométriques nous a permis d'identifier quatre caractéristiques spécifiques à chaque problème de placement : la représentation des obstacles, la forme géométrique à placer, les transformations considérées, et l'objectif de placement (qui dépend à la fois des obstacles et de la forme à placer).

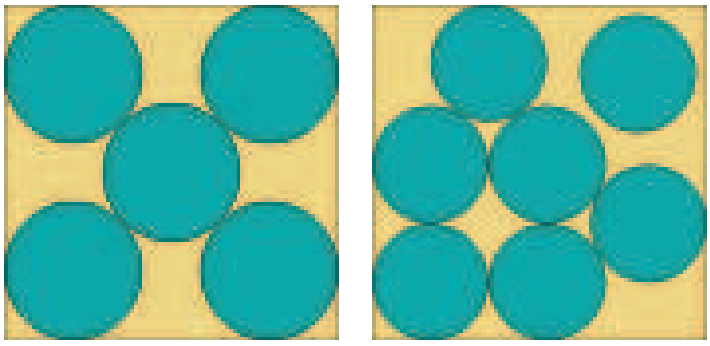
La sous-section suivante ouvre une parenthèse en faisant brièvement référence à quelques problèmes anciens appelés problèmes de *packing*. Nous avons jugé utile de les mentionner également au regard de notre analyse et d'expliquer en quoi ils diffèrent des problèmes de placement.

5.1.3 Placement et packing

Dans la sous-section précédente, on a vu que le *placement de formes géométriques* est un sujet étudié depuis les années 1980 environ, qui a donné lieu à une littérature abondante. Ces travaux publiés entretiennent un lien étroit avec une famille de problèmes qui est plus ancienne encore. Nous la mentionnons ici, du fait de son importance, tant historique qu'en termes de nombre de publications. En anglais, cette famille de problèmes directement issue des mathématiques et plus particulièrement des premières questions d'optimisation automatique est appelée *packing problems*. L'aspect historique est décrit dans l'état de l'art de Stephenson (2003).

Dans un problème de packing, il s'agit de « bien placer » un ensemble d'objets de forme connue dans un espace borné. Par exemple, il peut s'agir de déterminer le nombre maximal de disques de rayon 1 qu'il est possible de placer dans un carré dont le côté est un réel supérieur à quatre. Deux exemples de solution sont illustrées sur la figure 5.8. La simplicité de l'énoncé de tels problèmes (qui sont parfois utilisés comme simples casse-têtes dans les revues de jeux mathématiques) est trompeuse, plusieurs problèmes de packing demeurent en effet sans solution connue à ce jour, *cf.* article de Guy et al. (1994).

Un problème de packing peut être très complexe d'un point de vue calculatoire. C'est d'autant plus le cas que les objets à placer ont des formes



(a) *carré de côté* $2 + 2\sqrt{2}$ (b) *carré de côté* $4 + \sqrt{3}$

FIGURE 5.8 – *Exemples de solutions au problème du « circle packing in a square »*

variées. On sait par exemple, d'après l'article de Li et Milenkovic (1995) que le placement de polygones quelconques dans une zone rectangulaire donnée est un problème NP-difficile.

Du fait de cette difficulté intrinsèque, des heuristiques ont été proposées pour de tels problèmes. On en trouve des exemples dans l'article de : Chen et Huang (2009). Certains travaux, comme celui présenté dans l'article de Jakobs (1996), emploient même des procédés de calcul stochastiques avancés tels que les algorithmes évolutionnaires.

Les travaux présentés dans cette thèse ne sont pas directement liés au *packing*, aussi nous n'entrerons pas plus dans les détails. On considère qu'un problème de packing n'est pas un problème de « placement » au sens où nous l'avons présenté. Dans un problème de packing, chaque forme placée influence et contraint le placement des formes suivantes. Il y a là une différence majeure : dans le packing, les obstacles ou contraintes pour placer la forme ne sont pas toutes déterminées *a priori*.

Au terme de cette section nous avons donné un aperçu des travaux existants en *placement de formes géométriques*. En comparant ces travaux, nous avons identifié des caractéristiques spécifiques à chaque problème.

Nos propres recherches abordent le placement de formes géométriques où la forme à placer peut être *très complexe* : formes non convexes, pouvant contenir des trous, *etc.* Une telle forme est placée dans un nuage de points avec le souci d'obtenir des algorithmes efficaces et simples à mettre en œuvre. Dans la section qui suit, nous introduisons donc notre propre variante du placement de formes géométriques, pour aller plus loin dans l'étude de cette famille de problèmes.

5.2 Placement d'union de disques : définition et notations

Dans cette section nous présentons un cadre théorique permettant d'étudier différents problèmes de placement d'union de disques en utilisant un même formalisme. Ces problèmes sont des variantes sur le thème du placement de formes géométriques dont on a donné un aperçu dans la section précédente.

Pour faire le lien avec ce qui a été dit dans la sous-section 5.1.2 (com-

paraison des problèmes de placement), la variante que nous allons étudier modélise les obstacles via un nuage de points ; elle cherche à placer une union de disques quelconque et dans un premier temps on ne fait aucune restriction sur les transformations considérées. L'objectif est l'évitement des obstacles.

Ainsi, dans notre contexte nous parlerons uniquement de *placement d'union de disques dans un nuage de points*. Ci-dessous, nous donnons l'énoncé formel général de ce problème.

Les avantages d'une représentation par union de disques ont été mentionnés au chapitre 2. Nous avons aussi introduit un formalisme dédié. Notamment, la définition 2.2.3 introduisait la notion de *configuration* et la définition 2.2.4 celle de *copie paramétrée*. Rappelons qu'une *configuration* désigne un triplet de paramètres $a = (x, \lambda, \theta)$ pour une transformation combinant translation, homothétie, rotation. Les paramètres x, λ, θ sont respectivement le vecteur de la translation, λ le facteur d'homothétie et θ l'angle de rotation.

Dans toutes les définitions qui suivent, les symboles P et B désignent respectivement un nuage de points et un Ensemble de Disques Standard ou EDS *cf.* définition 2.2.1.

Définition 5.2.1 (**c**onfiguration autorisée)**.** Toute configuration a qui vérifie la relation : $\cup B(a) \cap P = \emptyset$ est appelée configuration autorisée.

Définition 5.2.2 (**placement d'union de disques dans un nuage de points).** Étant donné un ensemble E de configurations *i.e.* d'éléments de $\mathbb{R}^2 \times \mathbb{R}^+ \times [0, 2\pi[$, le *placement d'union de disques dans un nuage de points* consiste à déterminer toutes les configurations autorisées parmi E. Autrement dit, le problème associé à l'ensemble E consiste à calculer l'ensemble, noté $sol_{P,B}(E)$, de toutes les configurations autorisées membre de E. Formellement :

$$sol_{P,B}(E) := \{a \in E \mid \cup B(a) \cap P = \emptyset\}$$

Le problème de placement le plus général est illustré figure 5.9.

Dans la section qui suit, on donne quelques exemples d'espaces de configurations qui illustrent les notions que l'on vient d'introduire.

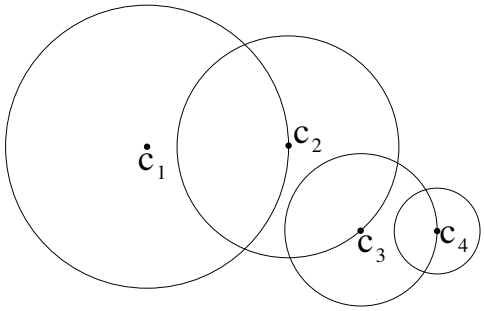
5.3 Espace de configurations

L'énoncé du placement d'union de disques dans un nuage de points a été donné par la définition 5.2.2. Cet énoncé a l'avantage d'être « paramétrable ». Dans cette section, via deux définitions nous allons introduire une terminologie portant sur le paramètre E utilisé. Ces deux définitions joueront un rôle essentiel dès le chapitre suivant.

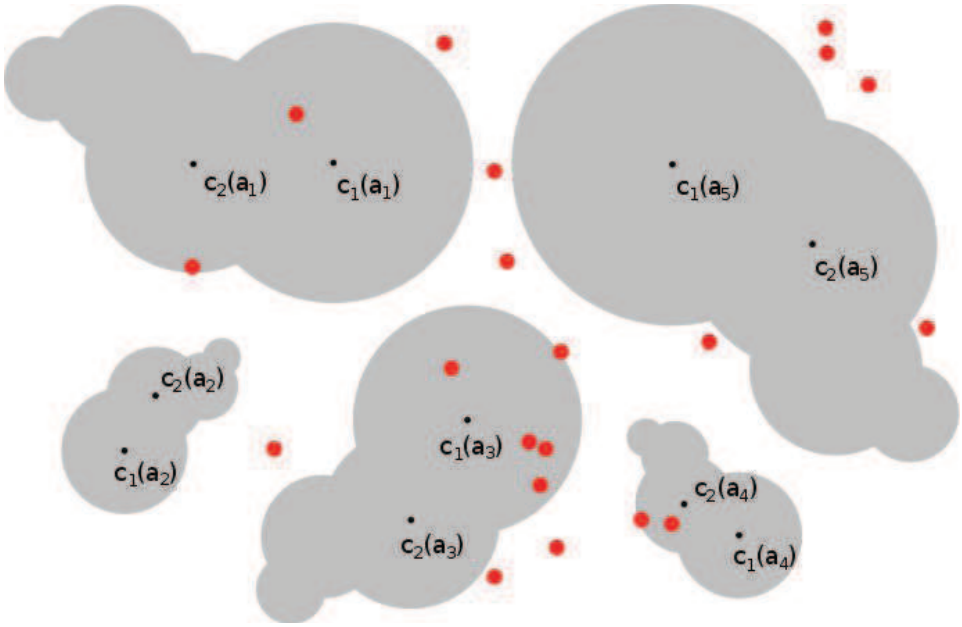
Définition 5.3.1 (**ensemble de toutes les configurations possibles).** L'ensemble de toutes les configurations possibles, noté Ω, est défini par le produit cartésien :

$$\Omega = \left(\mathbb{R}^2 \times \mathbb{R}^+ \times [0, 2\pi[\right)$$

Les ensembles $\mathbb{R}^2$, $\mathbb{R}^+$ et $[0, 2\pi[$ couvrent respectivement tous les paramètres possibles des transformations géométriques que sont : la translation, l'homothétie et la rotation.



(a) *Modèle de l'objet à placer*



(b) *Nuage de points considéré et cinq copies choisies arbitrairement*

Dans sa formulation la plus générale, le problème consiste à déterminer toutes les configurations permettant de placer un ensemble de disques (a) au sein d'un nuage de points, de sorte que l'union de ces disques ne contienne aucun point en son intérieur. Sur l'exemple (b), les points du nuage de points sont représentés en rouge. On a aussi représenté cinq copies choisies arbitrairement, ayant pour configurations respectives a_1, a_2, a_3, a_4, a_5. Parmi celles-ci, seules les configurations a_2 et a_5 font partie de l'ensemble-solution.

FIGURE 5.9 – *Problème du placement d'union de disques dans un nuage de points*

Comme on l'a déjà dit plus tôt dans ce chapitre lors de notre état de l'art, les problèmes de placement doivent faire un compromis entre efficacité algorithmique d'une part, et généralité de la solution d'autre part.

Le problème de placement le plus général consistant à calculer $sol_{P,B}(\Omega)$ est difficile. Sa résolution algorithmique, en supposant qu'elle soit possible, serait vraisemblablement très coûteuse en temps de calcul. Puisque l'énoncé est paramétrable, rien n'empêche de considérer un sous-ensemble de configurations, plutôt que toutes les configurations possibles. Ceci nous amène à la définition suivante.

Définition 5.3.2 (espace de configurations). On appelle espace de configurations et on note E tout sous-ensemble de l'ensemble Ω de toutes les configurations possibles. Un espace de configurations identifie un ensemble de transformations géométriques obtenues par une combinaison particulière de translation, homothétie et rotation.

Ainsi, tout espace de configurations vérifie la relation $E \subset \Omega$. Un espace de configurations permet de spécifier les transformations géométriques (ou opérations) applicables sur une union de disques $\cup B$ pour la placer. Ces transformations géométriques sont choisies parmi toute combinaison de translation, homothétie, rotation.

Dans la section suivante, grâce à cette définition que l'on vient de voir, nous allons énoncer explicitement et de manière cohérente tous les Problèmes de Placement (abrégés PP) d'union de disques traités durant nos travaux.

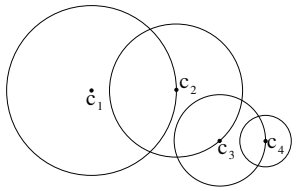
5.4 Problèmes de placement traités dans la thèse

Dans cette section, nous énonçons les PP étudiés au cours de la thèse. Pour chaque énoncé, on représente graphiquement le problème correspondant. Les problèmes énoncés sont au nombre de quatre : le placement par rotation uniquement, par homothétie uniquement, par homothétie-rotation combinées, par translation. La résolution de tous ces problèmes sera abordée au chapitre suivant.

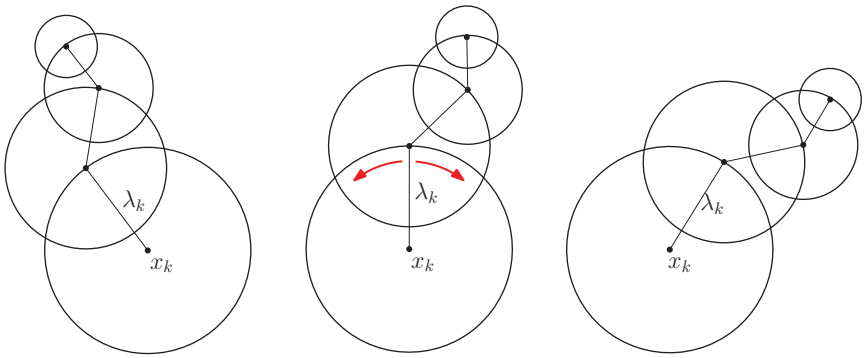
5.4.1 Placement par rotation uniquement

Le premier problème que nous avons abordé durant nos travaux est le placement par rotation. Du fait de la relative simplicité du problème, c'est cet exemple que l'on utilisera fréquemment pour illustrer les définitions abstraites données au chapitre suivant. Ci-dessous, nous en donnons une définition formelle. Dans ce qui suit, les symboles P et B désignent respectivement un nuage de points et un Ensemble de Disques Standard (EDS).

Définition 5.4.1 (PP par rotation uniquement). Soient x_k un point du plan et λ_k un réel positif fixés. Le PP par rotation uniquement est le problème identifié par l'espace de configurations $E_{R.}$ défini tel que : $E_{R.} = \{x_k\} \times \{\lambda_k\} \times [0, 2\pi[$ Ledit problème consiste donc à déterminer l'ensemble $sol_{P,B}(E_{R.})$ de toutes les configurations autorisées À partir de l'espace de configurations $E_{R.}$ et en tenant compte des données P, B.



(a) *EDS utilisé*



(b) *Trois exemples de copies*

FIGURE 5.10 – *Copies considérées par le* PP *par rotation*

Interprétation : Les configurations a membre de $E_{R.}$ sont toutes telles que la région $\cup B(a)$ est une rotation de l'union de disques $\cup B$. Pour le dire autrement, parmi toutes les copies $\cup B(a)$ pouvant être obtenues par rotation à partir de l'union $\cup B$, résoudre le PP par rotation consiste à déterminer quelles copies sont vides et lesquelles ne le sont pas.

Rappelons que le terme *copie* a été introduit au chapitre 2, via la définition 2.2.4.

Dans le cas du placement par rotation uniquement, les copies ont un même *point de référence* x_k, et une même *taille* λ_k. Elles diffèrent uniquement par leur orientation, comme illustré par la figure 5.10.

L'espace de configuration $E_{R.}$ détermine les copies $\cup B(a)$ à tester. Il s'agit de déterminer automatiquement lesquelles sont vides et lesquelles ne le sont pas.

Le placement par rotation d'une union de disques plus complexe est illustré sur la figure 5.11.

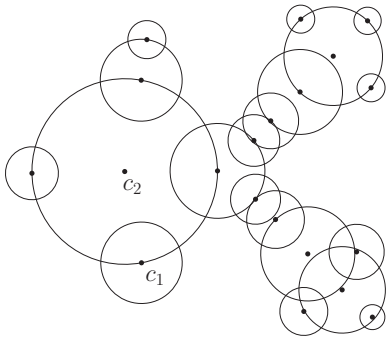
5.4.2 Placement par homothétie uniquement

Le principe du placement par homothétie est le même que pour le PP vu à la sous-section précédente, à une différence près : c'est la taille de la copie et non son orientation qui varie. L'énoncé formel est donné ci-dessous.

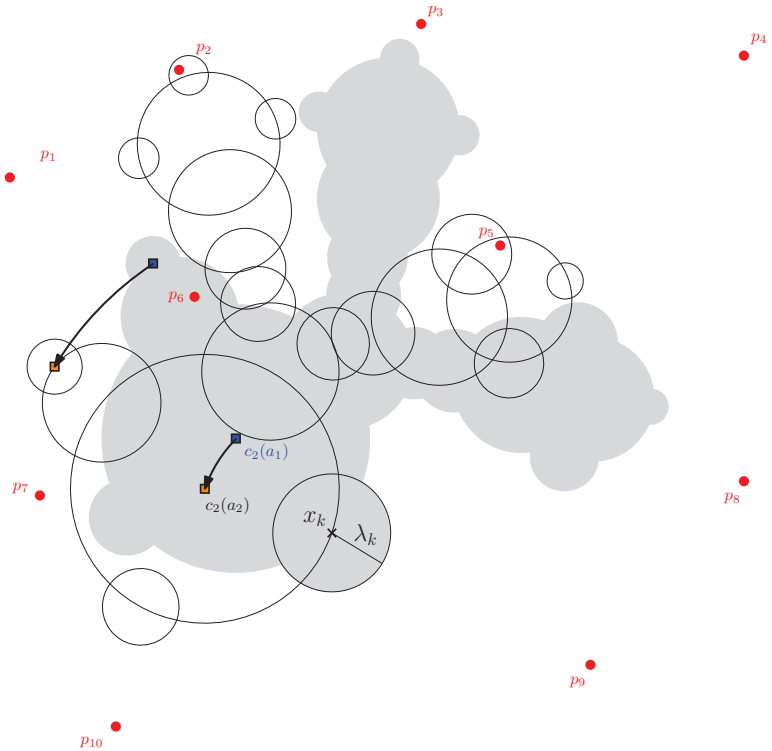
Définition 5.4.2 (PP par homothétie uniquement). Le placement d'une union de disques par homothétie uniquement correspond au calcul de l'ensemble des solutions $sol_{P,B}(E)$ dans le cas où $E = E_{H.}$ avec :

$$E_{H.} := \{x_k\} \times \mathbb{R}^+ \times \{\theta_k\}$$

Le centre $c_1(a)$ du premier disque est donc fixé, il est égal à x_k. L'orientation θ des copies considérées est elle aussi fixée, elle est égale à θ_k. Ces



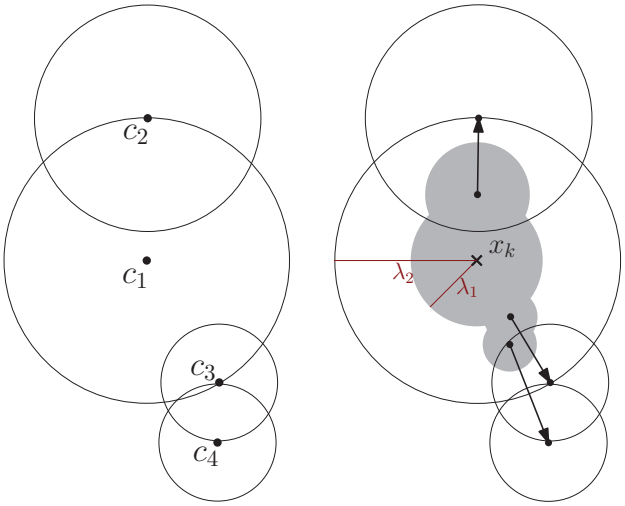
(a) *forme considérée pour le placement*



(b) *deux copies non-vides de la forme considérée. De fait, leur configuration n'appartient pas à l'espace des solutions. On voit que le passage d'une copie à l'autre correspond à une rotation (représentée par les deux flèches en gras) autour du point x_k (représenté par une croix)*

Pour ce problème, étant donnés une union de disques et un nuage de points, il s'agit de déterminer et de grouper toutes les copies telles que leur intersection avec le nuage de points soit vide

FIGURE 5.11 – *Placement d'union de disques par rotation*



(a) *EDS utilisé*

(b) *Deux exemples de copies, le passage de l'une à l'autre est réalisable par changement d'échelle par relativement à* x_k

FIGURE 5.12 – *Copies considérées par le* PP *par homothétie*

constantes $x_k \in \mathbb{R}^2, \theta_k \in \mathbb{R}^+$ constituent des données supplémentaires du problème.

Les copies considérées par le placement par homothétie sont représentées graphiquement 5.12, le problème lui-même est illustré par la figure 5.13.

5.4.3 Placement par homothétie et rotation combinées

L'espace de configurations $E_{R.H.}$ qui identifie la combinaison de la rotation et homothétie est : $E_{R.H.} = \{x_k\} \times \lambda \times [0, 2\pi[$ où x_k est un vecteur de translation fixé

Ce problème de placement combine les deux transformations vues précédemment : l'homothétie et la rotation.

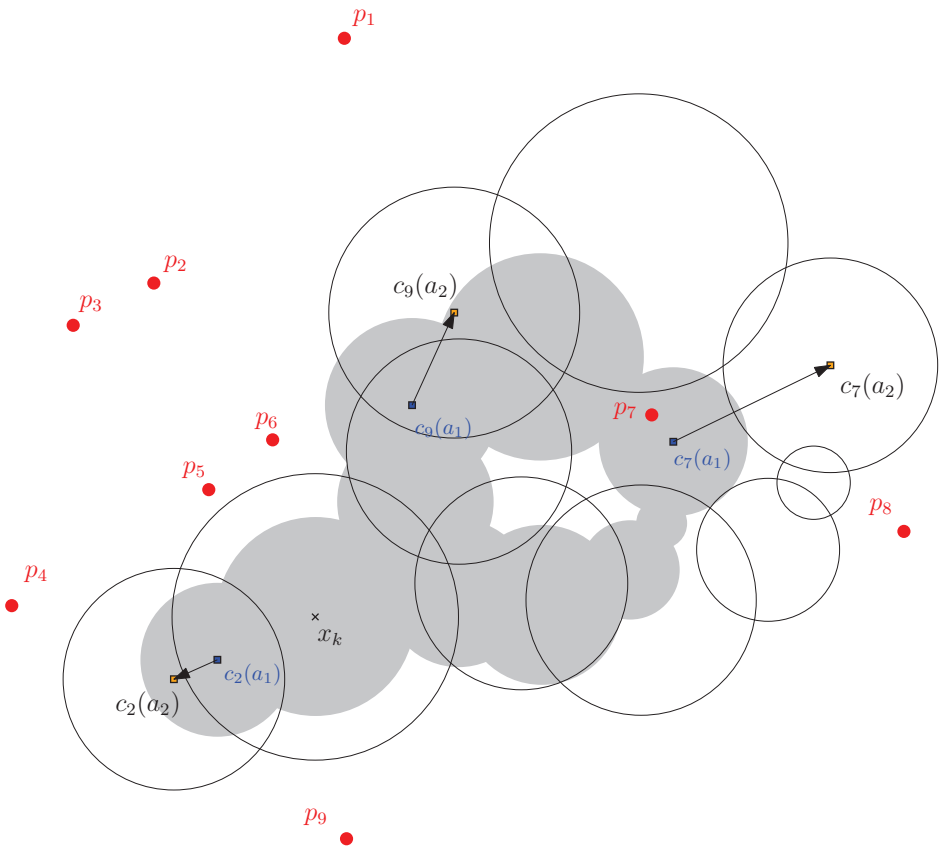
Définition 5.4.3 (PP par homothétie-rotation combinées). Le placement d'une union de disques par homothétie-rotation combinées correspond au calcul de l'ensemble des configurations autorisées, ensemble noté $sol_{P,B}(E)$, dans le cas où $E = E_{R.H.}$ avec :

$$E_{R.H.} := \{x_k\} \times \mathbb{R}^+ \times [0, 2\pi[$$

Pour toutes les copies considérées, le centre $c_1(a)$ du premier disque est fixé, il est égal à x_k qui constitue donc une donnée supplémentaire du problème.

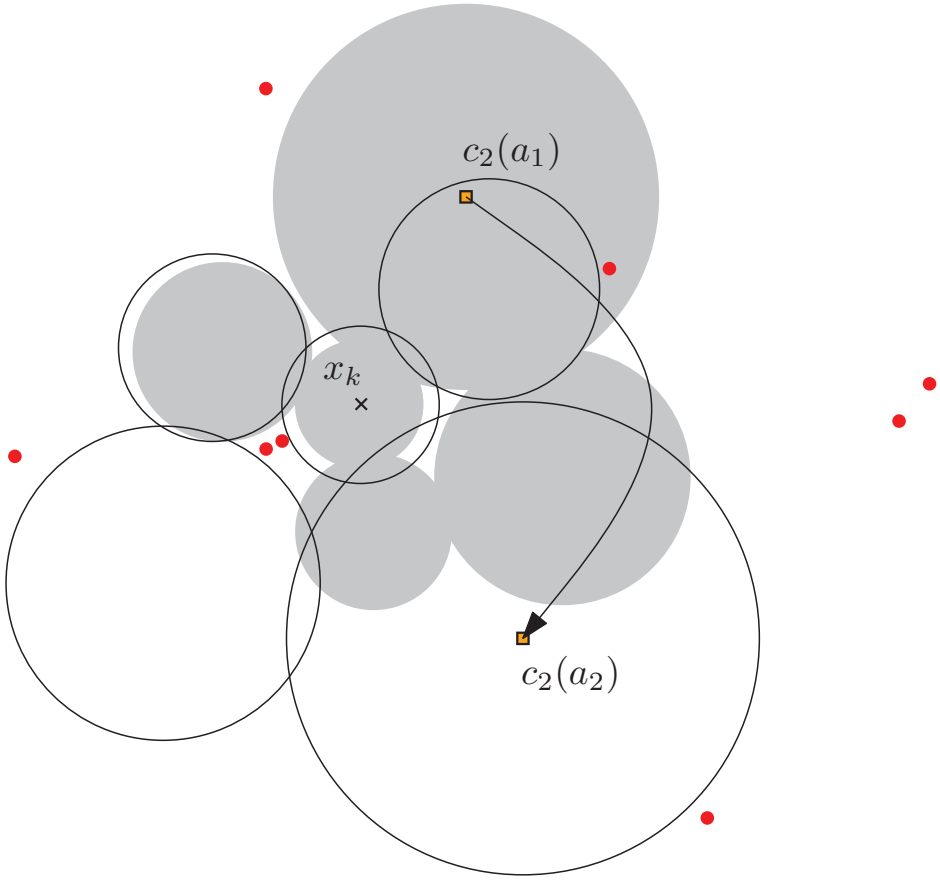
Ainsi dans ce cas, pour toutes les copies $\cup B(a)$ considérées durant le placement, ce n'est que la *taille de la copie* qui change.

Ce problème est illustré par la figure 5.14.



Les points de P sont représentés en rouge, deux copies considérées par le problème sont représentées pour l'exemple. Elles se trouvent dans des configurations a_1, a_2 respectivement. La copie $\cup B(a_1)$ n'est pas vide tandis que la copie $\cup B(a_2)$ l'est. Il s'agit de déterminer toutes les configurations a produisant des copies vides.

FIGURE 5.13 – *Problème de placement par homothétie*



Deux copies sont représentées pour l'exemple. Le passage de l'une à l'autre est possible par application d'une homothétie et rotation relativement au point x_k. Ainsi les configurations $a_1 = (x_k, \lambda_1, \theta_1)$ et $a_2 = (x_k, \lambda_2, \theta_2)$ ont seulement leur première composante x_k en commun. La copie $\cup B(a_2)$ est vide, contrairement à la copie $\cup B(a_1)$.

FIGURE 5.14 – *Problème de placement par homothétie et rotation combinées*

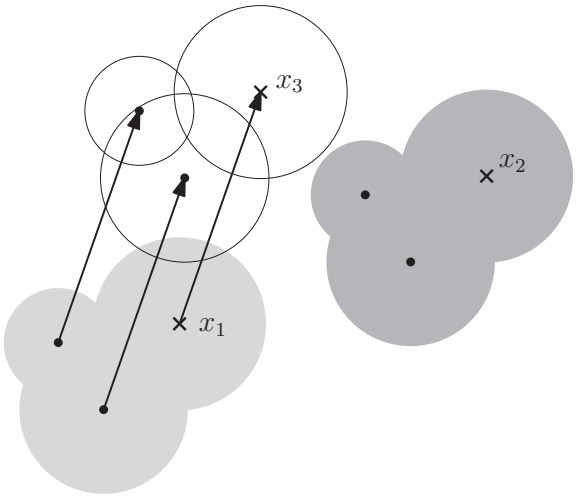


FIGURE 5.15 – *Copies considérées par le placement par translation. Le passage d'une copie à l'autre se fait en translatant simplement tous les disques par un même vecteur*

5.4.4 Placement par translation

Ci-dessous, comme pour les PP précédents, nous utilisons les notions nouvelles d'espace de configurations et de configurations autorisées pour définir le PP par translation.

Définition 5.4.4 (PP par translation). Le placement d'une union de disques par translation correspond au calcul de l'ensemble des configurations autorisées, ensemble noté $sol_{P,B}(E)$, dans le cas où $E = E_{T.}$ avec :

$$E = E_{T.} = \left(\mathbb{R}^2 \times \{\lambda_k\} \times \{\theta_k\}\right)$$

La taille de copie λ et l'orientation sont identiques pour toutes les copies considérées, elles sont respectivement égales à λ_k, θ_k. Ces constantes constituent une donnée supplémentaire du problème.

Le problème consiste donc à déterminer parmi les configurations de $E_{T.}$ toutes celles qui génèrent une copie vide. Les copies $\cup B(a)$ considérées par le PP par translation sont représentées graphiquement sur la figure 5.15.

Nous venons d'énoncer plusieurs problèmes de placement en utilisant un seul et même formalisme. Ce dernier est basé sur la notion d'espace de configurations. Ainsi, l'espace de configurations suffit à identifier sans ambiguïté tout problème de placement d'union de disques au sein d'un nuage de points. Dès lors, il devient possible de considérer ces problèmes de manière globale.

Avant de nous consacrer à la résolution de plusieurs d'entre eux (au prochain chapitre), nous traitons le cas particulier des espaces de configurations finis.

5.4.5 Cas particulier d'un espace de configuration fini

Rappelons qu'étant donnée une union de disques $\cup B$, une copie de celle-ci est notée $\cup B(a)$. On sait que toute copie est le résultat d'une certaine transformation géométrique (combinant éventuellement translation,

homothétie, rotation) appliquée à l'union de disques $\cup B$. Un PP dispose donc, selon l'espace de configurations E choisi, de possibilités de transformations géométriques plus ou moins nombreuses à appliquer sur l'union $\cup B$ pour placer celle-ci au sein du nuage de points P. Dans cette sous-section nous analysons le cas particulier où l'espace de configurations E serait de taille finie.

Lorsque l'espace de configurations E est fini, le PP se ramène au test d'un nombre fini de copies $\cup B(a)$. En procédant itérativement il est possible de déterminer, pour chaque configuration $a \in E$, si la condition $\cup B(a) \cap P = \emptyset$ est satisfaite. Plus précisément, on peut tester chaque couple $\big(p_j, b_i(a)\big) \in P \times B(a)$ un à un pour déterminer si le point p_j est contenu dans le disque $b_i(a)$. Ledit test se réduit à un simple calcul de distance entre le point p_j et le centre $c_i(a)$ suivi d'une comparaison au rayon $r_i(a)$ associé. Un algorithme « naïf » suffit donc à résoudre le problème, dans le cas où l'espace de configurations E est fini.

La complexité de cet algorithme peut être établie sans difficulté, donnons-en un aperçu. En notant k le nombre de configurations considérées, n le nombre de points de P et m le nombre de disques de B, l'algorithme résout le problème en temps $O(kmn)$. Pour tout cas où E est fini, le placement d'union de disques dans un nuage de points peut être résolu en temps polynomial.

La taille, ou plus précisément la *cardinalité*, de l'espace de configurations E est importante dans le mesure où elle donne une indication sur la difficulté à résoudre un PP. Remarquons que les problèmes que nous avons énoncé à la sous-section précédente considèrent un espace de configurations E de taille infinie. Autrement dit, pour chaque PP traité dans cette thèse, le nombre de placements à considérer est infini.

Conclusion du chapitre

Dans ce chapitre, après avoir présenté les travaux existants sur le placement, nous avons formalisé notre propre variante : le placement d'union de disques au sein d'un nuage de points. Ce faisant, nous avons notamment paramétré l'énoncé par un espace de configurations.

Il en résulte un cadre théorique puissant du fait de sa généralité : d'une part, le cadre peut représenter des formes géométriques à placer très diverses (grâce à l'union de disques). D'autre part, le fait d'avoir paramétré l'énoncé du problème par un espace de configurations permet de spécifier les opérations admises pour le placement. Le cadre global offre ainsi la possibilité d'exprimer et d'étudier, au moins de manière approchée, de la plupart des problèmes de placement. Notre démarche permet donc d'unifier la diversité qui existe dans l'étude du placement. Cette unification offre une comparaison bien plus directe des problèmes de placement entre eux, on peut aisément voir leurs différences et similitudes.

Au-delà de cet aspect pratique pour la réflexion, l'unification a des conséquences algorithmiques. Elle rend d'abord possible une classification des problèmes de placement. Et quel est intérêt à classifier les problèmes ? Cela va nous permettre, comme nous le verrons au chapitre suivant, de résoudre un problème de placement particulier et d'en généraliser la solution algorithmique à tout problème de sa classe.

6 Algorithme générique de placement de formes géométriques complexes

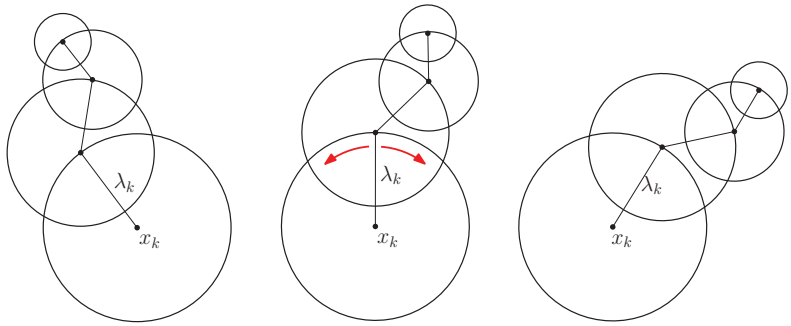
Sommaire

6.1 Résolution du placement par rotation 117
6.1.1 Rappels et identification d'un cas particulier 117
6.1.2 Définition d'orbite et d'arc interdit 118
6.1.3 Calcul de l'ensemble des configurations autorisées 120
6.2 Méthode générique de résolution des pp inversibles . . 125
6.2.1 Problèmes de placement inversibles 125
6.2.2 Généralisation des notions d'orbite et arc interdit 126
6.2.3 Résolution générique de tout pp inversible 128
6.2.4 Compactage des résultats 133
6.3 Exemples d'application de la méthode 135
6.3.1 Placement par homothétie uniquement 135
6.3.2 Placement par homothétie et rotation combinées 137
6.3.3 Placement par translation uniquement 140
Conclusion du chapitre 145

Où nous proposons un algorithme générique pour placer une forme géométrique complexe dans un nuage de points. Le terme « générique » signifie que la même solution algorithmique peut être utilisée pour résoudre différents problèmes de placement. L'utilisation *ad hoc* d'un paramètre (l'espace de configurations en l'occurrence) permet d'adapter la résolution à tel ou tel problème.

Dans un premier temps, nous présentons un algorithme itératif simple à comprendre et à implémenter pour résoudre le *placement par rotation*. L'algorithme en question fait usage de trois notions géométriques nouvellement définies : *orbites, arcs interdits*, et *fonction-centre réciproque*.

Dans un deuxième temps, nous généralisons les notions d'orbite, arc interdit et fonction-centre à tout problème membre d'une certaine classe, dite *classe des problèmes inversibles*. Pour l'exemple, nous appliquons l'algorithme de résolution obtenu à trois des problèmes énoncés au chapitre précédent, à savoir : placement d'union de disques par homothétie, par homothétie-rotation combinée et par translation.



Le placement d'union de disques par rotation dans un nuage de point considère les copies ayant un même premier centre $x_1 = x_k$ et une même taille $\lambda = \lambda_k$ fixés à l'avance. L'orientation des copies varie librement quant à elle.

FIGURE 6.1 – *Placement d'union de disques par rotation*

6.1 Résolution du placement par rotation

Au chapitre précédent, nous avons vu un cadre théorique pour énoncer divers problèmes de placement selon un même formalisme.

Dans cette section on s'intéresse à l'un d'entre eux en particulier : le placement par rotation, pour y apporter une solution concrète. La résolution de ce problème relativement simple sert d'introduction à une méthode plus puissante, mais aussi plus abstraite qui sera présentée dans la section suivante.

Ci-dessous, nous décrivons comment calculer des configurations autorisées, pour placer une union de disques dans un nuage de points par rotation uniquement. Notre exposé est structuré en trois parties : rappels et identification d'un cas particulier, définition illustrée des notions d'orbite et arc interdit, description pas à pas de l'algorithme. Chaque partie fait l'objet d'une sous-section dédiée.

6.1.1 Rappels et identification d'un cas particulier

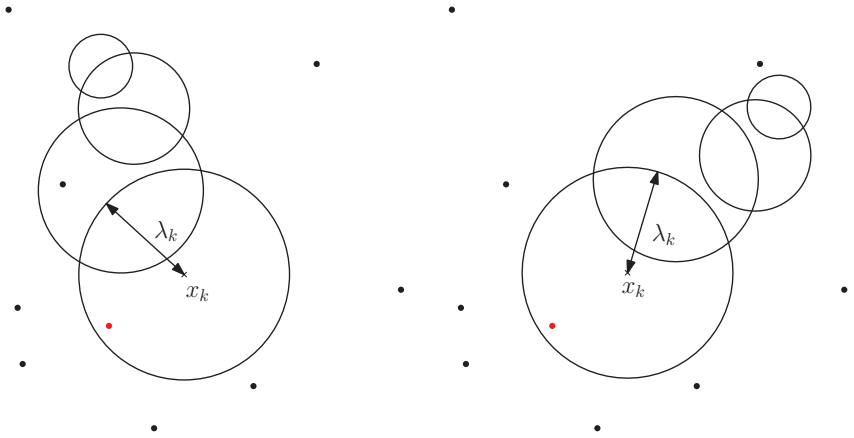
L'énoncé du problème de placement par rotation a déjà été donné définition 5.4.1. Nous le reformulons ici de manière un peu plus détaillée et faisons quelques rappels.

Rappels généraux sur le placement par rotation

Le placement d'union de disques par rotation prend en entrée un nuage de points P et un EDS noté B. De plus, comme on cherche à réaliser un placement par rotation uniquement, le premier centre est fixé : $c_1(a) = x_k$, de même que la taille des copies considérées $\lambda = \lambda_k$. L'orientation (paramètre θ) varie librement dans l'intervalle $[0, 2\pi[$. Le problème est illustré sur la figure 6.1.

En employant les définissions et notations présentées au chapitre précédent, le placement par rotation consiste à grouper toutes les configurations autorisées c'est-à-dire à calculer l'ensemble $sol_{P,B}(E_{R.})$ tel que (rappel de la définition 5.2.2) :

$$sol_{P,B}(E_{R.}) := \{a \in E_{R.} \mid \cup B(a) \cap P = \emptyset\}$$



(a) *un point (en rouge) fait que le premier disque n'est pas vide*

(b) *la variation d'orientation est sans influence sur cette observation*

Dans le contexte d'un placement d'union de disques par rotation, si le premier disque contient un point de P cela implique la résolution immédiate du problème : l'ensemble des solutions est égal à l'ensemble vide

FIGURE 6.2 – *Cas particulier pour le* PP *par rotation : non vacuité du premier disque*

La bonne compréhension de l'objectif du problème et des données en entrée met en évidence un cas particulier, cas pouvant être résolu directement.

Cas particulier : non-vacuité du premier disque des copies Il existe un cas particulier qui fait que la solution au placement par rotation est immédiatement connue. On l'a dit, le problème de placement par rotation suppose un point x_k fixé pour positionner les copies, et une taille λ_k constante elle aussi.

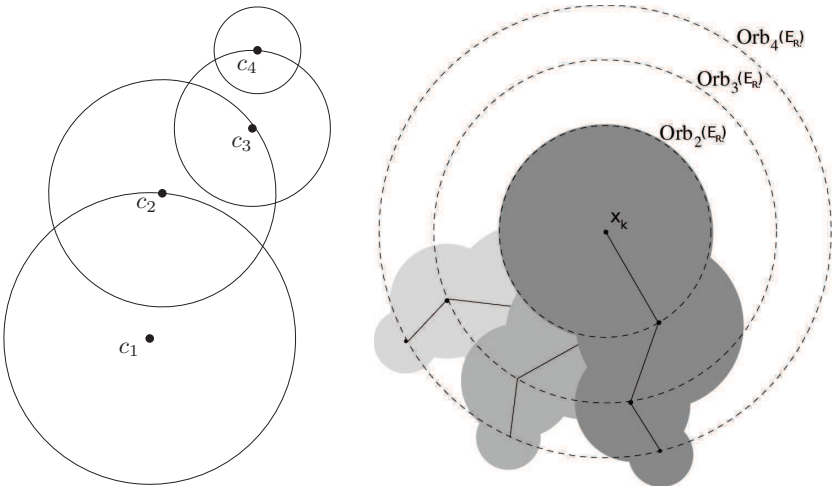
De fait, toutes les copies $\cup B(a)$ considérées partagent le même premier disque $b_1(a) = b(x_k, \lambda_k)$. Si ce dernier contient ne serait-ce qu'un point de P, aucune des copies considérées $\cup B(a), a \in E_{R.}$ ne peut être vide. Suivant ces hypothèses, il n'existe aucune configuration autorisée. Ainsi, l'ensemble des solutions est immédiatement connu : $sol_{P,B}(E_{R.}) = \emptyset$. Ce cas particulier est illustré par la figure 6.2.

Comme la résolution du problème est évidente dans ce cas particulier, l'étape d'initialisation de l'algorithme proposé plus loin consiste à vérifier si l'on se trouve dans ce cas. Pour le reste de l'algorithme, deux notions géométriques nouvelles sont pré-requises. Elle sont définies dans la sous-section suivante.

6.1.2 Définition d'orbite et d'arc interdit

Dans la présente sous-section, on introduit l'une après l'autre deux notions géométriques : l'orbite et l'arc interdit. Celles-ci seront utilisées par la suite pour calculer simplement les solutions du placement par rotation.

Premièrement, pour définir la notion d'orbite, on sait que pour le placement par rotation les paramètres x_k, λ_k sont fixés. Quelque soit la copie $\bigcup B(a), a \in E_{R.}$, la distance entre le premier et le i-ème



(a) *Ensemble de disques standard considéré pour cet exemple* (b) *Orbites associées une à une aux centres* $c_i(a)$

FIGURE 6.3 – *Orbite dans le contexte du placement par rotation*

L'orbite est une notion géométrique définie comme un ensemble de points tels que : $\forall a \in E_{R.}, c_i(a) \in orb_i(E_{R.})$.

centre est donc constante. Formellement, cela s'écrit : $\|c_1(a) - c_i(a)\| = \|x_k - (x_k + \lambda_k \mathbf{M}_\theta c_i)\| = \lambda_k \|c_i\|$. Cette valeur ne varie pas quelque soit la configuration $a \in E_{R.}$ que l'on considère. De ce constat découle la propriété géométrique énoncée ci-dessous.

Propriété 6.1.1. *Soit $a \in E_{R.}$ une configuration considérée par le* PP *par rotation. Le centre $c_i(a)$ du i-ème disque d'une copie $\bigcup B(a)$ se situe nécessairement sur le cercle centré en $c_1(a) = x_k$ et de rayon $\lambda_k \|c_i\|$. Formellement :*

$$\forall a \in E_{R.}, c_i(a) \in \partial\mathbf{b}\left(x_k, \lambda_k \|c_i\|\right)$$

Cette propriété est illustrée via la figure 6.3. Quelque soit la configuration $a \in E_{R.}$, le centre du i-ème disque d'une copie se situe toujours sur un même cercle. On va nommer *orbite* chacun de ces cercles. La définition est résumée ci-dessous.

Définition 6.1.2 (orbite). Soient B un EDS, x_k un point du plan, λ_k un réel positif. Pour $2 \leq i \leq m$ nous appelons i-ème orbite, et notons orb_i le cercle centré en x_k, de rayon $\lambda_k \|c_i\|$. Formellement :

$$orb_i(E_{R.}) := \partial\mathbf{b}\left(x_k, \lambda_k \|c_i\|\right) \text{ pour } 2 \leq i \leq m$$

Deuxièmement, pour définir la notion d'« arc interdit », on remarque que sur une orbite peuvent exister des arcs de cercle donnant une information sur le caractère vide ou non des disques composant les copies considérées. La définition de ces arcs de cercle particuliers faisant partie d'une orbite est donnée ci-dessous. Elle est illustrée par la figure 6.4.

Définition 6.1.3 (arc interdit). Étant donné B, x_k, λ_k comme précédemment et soit $i \in \mathbb{N}$, $2 \leq i \leq m$, $p_j \in P$, nous appelons arc interdit associé au couple i, p_j l'arc de cercle obtenu par intersection entre la i-ème orbite et le disque ouvert centré en p_j de rayon $\lambda_k r_i$. Formellement :

$$\delta_{i,p_j}(E_{R.}) := orb_i(E_{R.}) \cap b(p_j, \lambda_k r_i)$$

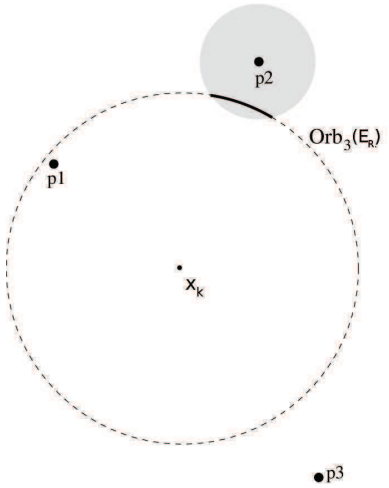


FIGURE 6.4 – *Exemple d'arc interdit associé à un point de P et à la troisième orbite*

Voyons une des conséquences de la définition d'arc interdit.

Propriété 6.1.4. *L'appartenance du point $c_i(a)$ (centre du i-ème disque d'une copie $\cup B(a)$) à l'arc interdit associé au couple b_i, p_j implique que le point p_j est contenu dans le disque b_i.*
Formellement :

$$c_i(a) \in \delta_{i,p_j}(E_{R.}) \Rightarrow p_j \in b_i(a)$$

Démonstration.

$$\begin{aligned} c_i(a) \in \delta_{i,p_j} &\Leftrightarrow c_i(a) \in orb_i(E_{R.}) \cap b(p_j, \lambda_k r_i) \\ \text{Or, } c_i(a) \in orb_i(E_{R.}) \cap b(p_j, \lambda_k r_i) &\Rightarrow \left\| c_i(a) - p_j \right\| < \lambda_k r_i \\ &\Rightarrow p_j \in b(c_i(a), \lambda_k r_i) \\ &\Rightarrow p_j \in b_i(a) \end{aligned}$$

□

Nous venons d'introduire les notions d'orbite et d'arc interdit.

Celles-ci vont nous permettre de résoudre le placement par rotation, c'est-à-dire de calculer toutes les configurations autorisées pour ce problème précis. L'algorithme utilisant ces notions est présenté dans la sous-section suivante.

6.1.3 Calcul de l'ensemble des configurations autorisées

En partant de l'ensemble des configurations considérées pour le placement par rotation, comment calculer l'ensemble des configurations autorisées ? Nous allons répondre à cette interrogation étape par étape, pour aboutir à un algorithme. Le nombre de vecteurs de l'espace de configuration $E_{R.}$ étant infini, les notions introduites précédemment (orbite et arc interdit) vont nous être grandement utiles. Elles sont intéressantes dans la mesure où elles fournissent des informations sur la position des centres des disques, ainsi que sur le caractère vide ou non de ces disques par rapport à P, pour toutes les configurations considérées. On utilise également une fonction particulière que l'on introduit en premier lieu.

Fonction-centre réciproque

Au chapitre 2, nous avons introduit la notation c_i pour désigner le centre du i-ème disque d'un EDS noté B. En définissant la notion de *copie* d'une union de disques, nous avons introduit la notation $B(a)$ pour signifier que les coordonnées des centres et les rayons sont modifiés selon une configuration a. Par là même, nous avons autorisé un abus de notation en laissant c_i désigner une fonction qui à tout a associe de nouvelles coordonnées.

Les fonctions-centre $\mathbf{c_i}$ ont été introduites dans la définition 2.2.5. On rappelle qu'une telle fonction associe à toute configuration $a = (x, \lambda, \theta)$, le point du plan $c_i(a) = x + \lambda \mathbf{M}_\theta c_i$ qui est en fait le centre du i-ème disque de la copie $B(a)$. Ci-dessous nous énonçons une propriété supplémentaire associée à ces fonctions, propriété qui va nous servir à résoudre le problème.

Propriété 6.1.5. *En restreignant les fonctions-centre c_i à l'espace de configurations $E_{R.}$, pour tout $2 \leq i \leq m$, il existe une fonction réciproque c_i^{-1} qui est la réciproque de c_i.*

Démonstration. En restreignant les fonctions-centre c_i à l'espace de configurations $E_{R.}$, la définition des fonctions c_i devient :

$$\begin{array}{rccl} c_i : & E_{R.} & \rightarrow & \mathbb{R}^2 \\ & a = (x_k, \lambda_k, \theta) & \mapsto & x_k + \lambda_k . \mathbf{M}_\theta . c_i \end{array}$$

Ci-dessous, pour tout point du plan z, on note $\angle(z)$ l'angle orienté formé par le vecteur $\overrightarrow{\mathcal{O}z}$ et le vecteur unitaire $\overrightarrow{\mathcal{O}X}$ parallèle à l'axe des abscisses. On rappelle que le symbole $\mathcal{O}$ désigne l'origine du repère cartésien. Nous introduisons une suite de fonctions g_i, définie pour $2 \leq i \leq m$. À tout point du plan, une fonction g_i associe un triplet de $E_{R.}$ tel que :

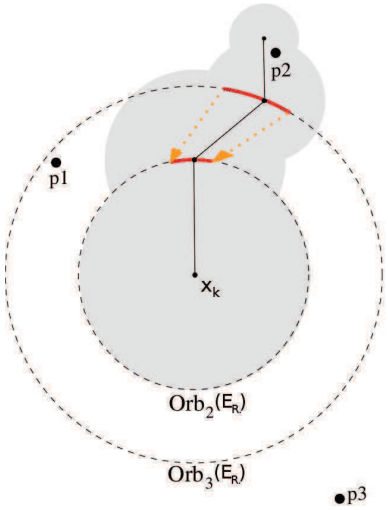
$$\begin{array}{rccl} g_i : & \mathbb{R}^2 & \rightarrow & E_{R.} \\ & z & \mapsto & (x_k, \lambda_k, \angle(z - x_k) - \angle(c_i)) \end{array}$$

Les fonctions g_i ainsi définies sont en fait les fonctions une à une réciproques de c_i. Vérifions-le en calculant la composée des deux fonctions g_i et c_i (composée notée $g_i \circ c_i$ avec les notations mathématiques habituelles). Soit $a = (x_k, \lambda_k, t)$ une configuration quelconque de $E_{R.}$. Nous avons :

$$\begin{array}{rcl} g_i(c_i(a)) & = & g_i\left(x_k + \lambda_k . \mathbf{M}_t . c_i\right) \\ & = & (x_k, \lambda_k, \angle(x_k + \lambda_k . \mathbf{M}_t . c_i - x_k) - \angle(c_i)) \\ & = & (x_k, \lambda_k, \angle(\lambda_k . \mathbf{M}_t . c_i) - \angle(c_i)) \\ & = & (x_k, \lambda_k, \angle(c_i) + t - \angle(c_i)) \\ & = & (x_k, \lambda_k, t) = a \end{array}$$

Quelque soit i compris entre 2 et m, la composée $g_i \circ c_i$ est égale à la fonction identité. Donc la fonction g_i est bien la réciproque de la fonction-centre c_i. Dès lors, on utilise la notation mathématique habituelle c_i^{-1} pour désigner la fonction réciproque de c_i. □

On appellera *problème de placement inversible* tout problème de placement d'union de disques pour lequel la fonction-centre c_i admet une fonction réciproque g_i. Le placement par rotation est donc un problème inversible.



La fonction-centre réciproque permet de calculer l'ensemble des configurations associées à un ensemble de points situés sur une orbite donnée. Ceci permet, connaissant un arc situé sur la troisième orbite, de calculer son « équivalent »sur une autre orbite

FIGURE 6.5 – *Fonction-centre réciproque dans le contexte du placement par rotation*

De plus, les fonctions-centre c_i étant bijectives dans le cas d'un problème inversible, elles peuvent être appliquées sur des sous-ensembles de leur domaine de définition sans perte d'information. Cette idée est illustrée par la figure 6.5.

Algorithme pour le placement par rotation

L'algorithme de placement par rotation groupe toutes les configurations autorisées pour placer la forme (union de disques) sans qu'elle contienne un seul point de P en son intérieur. On calcule ainsi, à partir de l'espace de configurations $E_{R.}$, toutes les configurations a telles que $\cup B(a) \cap P = \emptyset$.

Si (il existe $p \in P, p \in b(x_k, \lambda_k)$) **Alors**
  retourner $\emptyset$ l'ensemble vide
Fin Si
initialiser $tmpAI$ une variable représentant un arc interdit
initialiser $listeCNV$ une liste de configurations non autorisées
Pour i **de** 2 **à** m **faire**
  Pour j **de** 1 **à** n **faire**
    calculer arc interdit $\delta_{i,p_j}(E_{R.})$, le stocker dans $tmpAI$
    calculer l'image $c_i^{-1}(tmpAI)$ et l'ajouter à $listeCNA$
  Fin Pour
Fin Pour
Retourner la différence ensembliste $E_{R.} \setminus listeCNA$

Algorithme 5 – *résolution du* PP *par rotation*

L'algorithme 5 décrit les étapes de calcul en langage pseudo-code. *Grosso modo*, l'ensemble des solutions, noté $sol_{P,B}(E_{R.})$, est déterminé par élimination. On calcule au fur et à mesure les configurations non-autorisées, pour aboutir par différence à la solution recherchée. Détaillons ce principe de résolution qui suit quatre grandes étapes :

Etape 1. On vérifie si l'on se trouve dans le cas particulier énoncé dans la sous-section 6.1.1. Si c'est le cas, le problème est immédiatement résolu. Il n'existe pas de configuration autorisée.

Etape 2. On démarre une boucle où pour chaque couple : $b_i \in B$, $p_j \in P$ on calcule l'arc interdit associé. Concrètement, il suffit de calculer l'intersection entre l'orbite $orb_i(E_{R.})$ *i.e.* le cercle $\partial b(x_k, \lambda_k \|c_i\|)$ et un disque ouvert centré en p_j, comme expliqué dans la définition 6.1.3. Le calcul d'arc interdit est l'étape-clé du procédé. En effet, tout arc interdit fournit l'information suivante :

$$c_i(a) \in \delta_{i,p_j}(E_{R.}) \Rightarrow p_j \in b_i(a)$$

Etape 3. À chaque étape de la boucle, on applique également la fonction-centre réciproque c_i^{-1} sur l'arc interdit pour déterminer les configurations associées à ces ensembles de points. La fonction-centre réciproque c_i^{-1} permet ainsi d'identifier les configurations qui produisent ces disques dont le centre se trouve sur un arc interdit.

$$a \in c_i^{-1}(\delta_{i,p_j}(E_{R.})) \Rightarrow c_i(a) \in \delta_{i,p_j}(E_{R.})$$

La bijectivité de la fonction c_i permet de déterminer les angles (par extension : les configurations) correspondants aux arcs interdits. On regroupe une information jusqu'alors dispersée sur des orbites distinctes, l'application de la fonction c_i^{-1} prolonge ainsi la propriété des arcs interdits ; elle donne :

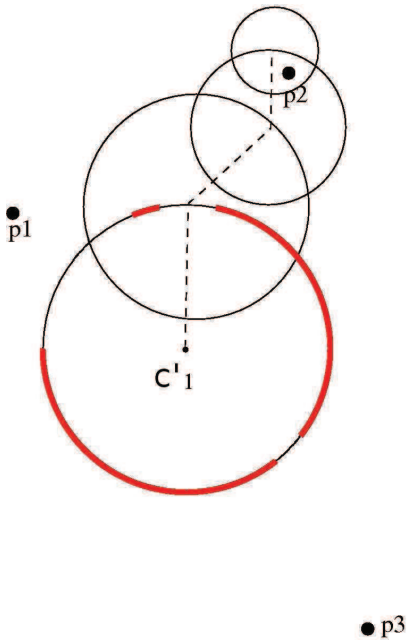
$$a \in c_i^{-1}(\delta_{i,p_j}(E_{R.})) \Rightarrow a \notin sol_{P,B}(E_{R.})$$

Etape 4. Une fois les images de tous les arcs interdits $\delta_{i,p_j}(E_{R.})$ calculées par la fonction c_i^{-1}, elles sont stockées dans une variable $listeCI$. Il suffit de retourner la différence de $E_{R.} \setminus listeCI$.

En calculant tous les arcs interdits et en leur appliquant la fonction-centre réciproque c_i^{-1}, l'ensemble de solutions $sol_{P,B}(E_{R.})$ a bien été déterminé par élimination. Une preuve détaillée de la validité de l'algorithme sera donnée dans la section suivante, dans un cadre plus général.

Pour représenter l'ensemble de solutions, nous avons tracé et mis en évidence $c_2(sol_{P,B}(E_{R.}))$. Comme toutes les formes considérées ont leur second centre $c_2(a)$ situé sur le cercle $\partial b(x_k, \lambda_k \|c_2\|)$, l'image de la fonction-centre c_2 appliquée aux solutions consiste en un ensemble d'arcs de cercle. Ainsi, il faut et il suffit que le point $c_2(a)$ fasse partie de cet ensemble d'arcs de cercle pour que $a \in sol_{P,B}(E_{R.})$ c'est-à-dire que $\cup B(a)$ soit vide par rapport à P. La figure 6.6 illustre le résultat de l'algorithme 5.

On vient de présenter un algorithme qui résout le placement par rotation. La section suivante généralise cette approche à d'autres problèmes de placement, tout en gardant la simplicité de son implémentation.



Pour cet exemple, l'ensemble de disques B utilisé est composé de 4 disques, le nuage de points consiste simplement en trois points $P = \{p_1, p_2, p_3\}$. Les arcs de cercle représentés en gras désignent les configurations autorisées (configurations telles que la copie ne contienne pas de point en son intérieur). Toute la copie $\cup B(a)$ est vide si et seulement si son second centre $c_2(a)$ se trouve sur l'un de ces arcs. La copie illustrée n'est pas vide et on voit effectivement que son centre $c_2(a)$ n'appartient pas aux arcs en gras.

FIGURE 6.6 – *Solution calculée pour une instance du placement par rotation*

6.2 Méthode générique de résolution des PP inversibles

Dans la section précédente, nous avons présenté l'algorithme 5 qui résout le placement d'union de disques par rotation.

L'objet de la présente section est d'expliquer comment nous avons généralisé ce premier résultat, dans le but d'obtenir une méthode de résolution *générique*. La nouvelle méthode est générique au sens où elle fonctionne indifféremment sur divers problèmes de placement.

La présente section est structurée en quatre sous-sections : Les deux premières (respectivement intitulées problèmes de placement inversibles, définitions géométriques généralisées) décrivent le moyen par lequel l'extension de l'algorithme 5 est possible. La troisième présente l'algorithme générique, apte à résoudre tout problème de placement inversible. À ce niveau, on donnera la preuve de sa validité et une étude de complexité algorithmique. La quatrième présente une analyse des résultats produits par l'algorithme, en considérant l'espace mémoire qu'occupent ces résultats.

6.2.1 Problèmes de placement inversibles

Une des caractéristiques du placement par rotation, on l'a déjà vu, est l'existence de fonctions réciproques c_i^{-1}. Pour généraliser l'algorithme 5 (algorithme du placement par rotation) à d'autres problèmes, ceux-ci doivent disposer de cette même caractéristique.

Définition 6.2.1 (problème de placement inversible). Soit E un espace de configurations. Le problème de placement identifié par E est dit inversible si et seulement si la fonction-centre c_i restreinte à E admet une fonction réciproque c_i^{-1}, pour tout $2 \leq i \leq m$.

Remarque. Tous les problèmes de placement (PP) ne sont pas des problèmes inversibles. Il n'est pas toujours possible de définir une famille de fonctions c_i^{-1} réciproques. Prenons comme exemple le PP utilisant translation et rotation combinées. Ce problème est identifié par l'espace de configurations suivant :

$$E_{R.T.} := \mathbb{R}^2 \times \{\lambda_k\} \times [0, 2\pi[\text{ où } \lambda_k \text{ est un réel positif fixé}$$

Dans ce contexte, il existe deux configurations distinctes $a, a' \in E_{R.T.}$ dont l'image par la fonction-centre est identique. Formellement : $a \neq a'$ et $c_i(a) = c_i(a')$. Pour s'en persuader, le lecteur peut considérer un ensemble de disques simple : $B = \{b(\mathcal{O}, 1), b((1,0), 1)\}$ par exemple, et calculer les images de $a = ((0,0), 1, 0)$ et de $a' = ((1,0), 1, \pi)$. Dans un tel cas il existe donc un point de $\mathbb{R}^2$ qui possède plus d'un antécédent dans $E_{R.T.}$, par conséquent la fonction-centre restreinte au domaine $E_{R.T.}$ n'est pas injective. Et *a fortiori* cette fonction-centre n'est pas bijective. Le problème pour lequel $E = E_{R.T.}$ n'est pas un problème inversible.

Exemple 3. Comme cela a déjà été énoncé par la propriété 6.1.5, dans le contexte du placement par rotation d'une union de disques dans un nuage de points, on dispose effectivement d'une fonction-centre réciproque. De fait, le placement par rotation uniquement est un PP inversible.

Remarque. Cette définition a une conséquence importante : si un PP est inversible alors il existe aussi une bijection entre son espace de configuration E, et une région du plan euclidien $\mathbb{R}^2$. Par extension, il existe une bijection entre l'ensemble des solutions $sol_{P,B}(E) \subset E$ et une partie du plan euclidien. Cela constitue alors un avantage pratique, dans la mesure où l'on peut *représenter graphiquement* les solutions d'un PP inversible en traçant des régions.

Ainsi, tout problème inversible est semblable au placement par rotation, en ceci qu'il dispose également d'une fonction-centre réciproque. L'ensemble de ceux-là forment ce que nous appelons dorénavant *classe des problèmes inversibles*.

Dans cette sous-section nous avons introduit la notion d'*inversibilité* d'un problème de placement d'union de disques. L'inversibilité ou la non-inversibilité permettent de *classer* les problèmes de placement. On va voir qu'il est possible d'étendre les notions d'orbite et d'arc interdit à toute la classe des problèmes inversibles.

6.2.2 Généralisation des notions d'orbite et arc interdit

Rappelons que le procédé de résolution du placement rotation présenté plus tôt utilise trois notions géométriques nouvelles. Celles-ci sont désignées respectivement par les termes : orbite, arc interdit et fonction-centre réciproque.

Dans la sous-section précédente, nous avons identifié les problèmes de placement qui sont « semblables » au placement par rotation, il s'agit des *problèmes inversibles*. Cette distinction constituait une première étape pour généraliser le procédé de calcul de l'algorithme 5.

La deuxième étape a consisté à étendre les notions d'orbite et arc interdit pour qu'elles s'adaptent à tout problème inversible.

Dans cette sous-section, nous réécrivons donc ces définitions dans un contexte plus général, et les renommons *orbite générale* et *zone interdite*. Commençons par l'orbite générale.

Définition 6.2.2 (orbite générale). Soit E l'espace de configurations considéré par un PP inversible. Pour $2 \leq i \leq m$, la i-ème orbite générale notée $orb_i(E)$ est le lieu du centre $c_i(a)$, pour une configuration $a \in E$ choisie telle que le disque $b_1(a)$ soit vide de P. Formellement :

$$orb_i(E) = \left\{ y \in \mathbb{R}^2 \mid \exists a \in E; y = c_i(a) \text{ et } b_1(a) \cap P = \emptyset \right\}$$

Attention à bien distinguer les termes *orbite générale* et *orbite*. Le premier terme désigne une notion qui généralise la notion d'orbite. La notion d'orbite est, quant à elle, plus précise (elle est graphiquement représentable) et destinée au seul placement par rotation.

Exemple 4. Vérifions que la définition 6.1.2 d'orbite vue pour le placement par rotation est bien un cas particulier de la définition 6.2.2 donnée ci-dessus. Voyons les conséquences de l'hypothèse $E = E_{R.}$ sur la définition précédente :

$$orb_i(E_{R.}) = \left\{ y \in \mathbb{R}^2 \mid \exists a \in E_{R.}; y = c_i(a) \text{ et } b_1(a) \cap P = \emptyset \right\} \tag{6.1}$$

Dans le contexte du placement par rotation, nous avions énoncé une condition sur les valeurs de x_k, λ_k pour que le calcul de placement soit utile. Il s'agissait de la condition : $\|p - x_k\| \geq \lambda_k, \forall p \in P$, de sorte que $\forall a \in E_{R.}, b_1(a) \cap P = \emptyset$. Lorsque cette même condition est satisfaite, l'équation 6.1 devient :

$$\begin{aligned} orb_i(E_{R.}) &= \left\{ y \in \mathbb{R}^2 \mid \exists a \in E_{R.}; y = c_i(a) \right\} \\ &= c_i(E_{R.}) \\ &= \{ y \in \mathbb{R}^2 \mid y = x_k + \lambda_k . \mathbf{M}_\theta c_i \text{ avec } \theta \in [0, 2\pi[\} \\ &= \partial b \left(x_k, \lambda_k \left\| c_i \right\| \right) \end{aligned}$$

À partir de la définition 6.2.2, on « retombe » bien sur la définition d'orbite comme présentée pour le cas de la rotation. Ainsi, les deux définitions sont *compatibles* en ceci que l'une est la généralisation de l'autre.

En considérant un sous-ensemble de l'espace de configurations E, et en reprenant la fonction-centre c_i, nous avons mis en évidence la propriété générale suivante.

Propriété 6.2.3. *L'orbite générale $orb_i(E)$ est égale à l'image par la fonction-centre c_i du sous-ensemble $E' \subset E$ défini par : $E' = \{a \in E \mid b_1(a) \cap P = \emptyset\}$ Formellement :*

$$orb_i(E) = c_i\left(E'\right)$$

Démonstration. En suivant un raisonnement analogue à celui de l'exemple donné ci-dessus et du fait de la définition de E' la démonstration est immédiate. □

Comme cela a été fait pour l'orbite, nous avons généralisé la notion d'arc interdit pour aboutir à la définition ci-dessous, valable pour tout PP inversible.

Définition 6.2.4 (zone interdite)**.** Soit E l'espace de configurations considéré par un PP inversible. Pour $2 \leq i \leq m$ la i-ème zone interdite, notée $\Delta_i(E)$, est définie comme étant le sous-ensemble de la i-ème orbite tel que pour tout point z lui appartenant, le i-ème disque de l'unique copie $\cup B(a)$ telle que $z = c_i(a)$ n'est pas vide par rapport à P. Formellement :

$$\Delta_i(E) := \left\{ z \in orb_i(E) \mid b_i(a) \cap P \neq \emptyset, a = c_i^{-1}(z) \right\}$$

En considérant un point particulier p_j élément de P, la zone interdite peut être exprimée comme l'union de zones interdites *élémentaires*, comme défini ci-dessous.

Définition 6.2.5 (zone interdite élémentaire). Soit E l'espace de configurations considéré par un PP inversible. Pour i allant de 2 à m, étant donné un point $p_j \in P$ on appelle *zone interdite élémentaire* associée au couple (i, p_j) et on note $\delta_{i,p_j}(E)$, le sous-ensemble de la i-ème orbite $orb_i(E)$ défini par :

$$\delta_{i,p_j}(E) := \left\{ x \in orb_i(E) \mid p_j \in b_i(a) \text{ où } a = c_i^{-1}(x) \right\}$$

Les définitions 6.2.4 et 6.2.5 sont très proches. La propriété qui suit présente l'équation qui lie simplement zone interdite et zone interdite élémentaire.

Propriété 6.2.6. *La i-ème zone interdite est égale à l'union de toutes les zones interdites élémentaires associées à i, p_j pour tout point p_j de P. Formellement :*

$$\Delta_i(E) = \bigcup_{p_j \in P} \delta_{i,p_j}(E)$$

La généralisation des deux notions géométriques utilisées pour résoudre le placement par rotation vient d'être présentée.

La notion d'*orbite générale* englobe celle d'orbite, la notion de *zone interdite* englobe celle d'arc interdit. De plus nous avons vu qu'une zone interdite est une région qui peut se décomposer en *zones interdites élémentaires*. À présent, on dispose de tous les outils pour reprendre l'algorithme opérant sur le placement par rotation et le transformer en une méthode générique. Nous l'expliquons dans la sous-section suivante.

6.2.3 Résolution générique de tout pp inversible

La résolution du placement par rotation a été présentée à la section 6.1. La méthode proposée était basée sur trois notions : fonction-centre réciproque, orbite, arc interdit. Dans les deux sous-sections précédentes, ces mêmes notions ont été généralisées à tout PP inversible.

En conséquence, l'algorithme 5 se généralise aussi, tout en conservant sa simplicité, permettant de résoudre tout PP inversible.

La présente sous-section a pour objet la description de ce nouvel algorithme, qui est *générique* au sens où il résout indifféremment tout *problème inversible*. L'exposé est structuré en trois parties : énoncé en pseudo-code, validité de l'algorithme, et analyse de complexité.

Algorithme générique résolvant tout problème inversible

$E' \leftarrow$ calcul $\{a \in E \mid b_1(a) \cap P = \emptyset\}$ //premier disque vide
Si $(E' = \emptyset)$ **Alors**
  retourner $\emptyset$ l'ensemble vide
Fin Si
initialiser $tmpZI$ une variable codant une zone interdite.
initialiser $tabImgZI$ une variable pouvant coder une liste de zones interdites.
Pour i **de** 2 **à** m **faire**
  Pour j **de** 1 **à** n **faire**
    $tmpZI \leftarrow$ calculer zone interdite élémentaire $\delta_{i,p_j}(E)$.
    $tabImgZI$: ajouter $c_i^{-1}(tmpZI)$ l'image par fonction-centre réciproque.
  Fin Pour
Fin Pour
Retourner $E' \setminus \bigcup tabImgZI$ le complément de l'union des tous les ensembles calculés
permet de résoudre, en plus du placement par rotation, n'importe quel autre problème inversible

Algorithme 6 – *Algorithme générique résolvant tout* PP *inversible*

L'algorithme 6 décrit ci-dessus en pseudo-code décrit la résolution d'un problème inversible, quel qu'il soit.

Remarque. La solution retournée peut aussi être calculée via l'opération de *complément ensembliste* et un ensemble $A = \{a \in E \mid b_1(a) \cap P \neq \emptyset\}$ qui est le dual de E'. En effet,

$$\complement_E \left(\bigcup \{A, tabImgZI\} \right) = E' \setminus \bigcup tabImgZI$$

Rappelons quelles sont les données en entrée/sortie pour l'algorithme générique. Les données en entrée sont :

- P un nuage de n points du plan euclidien
- B un ensemble standard de m disques de $\mathbb{R}^2$
- $E \subsetneq \Omega$ un ensemble de configurations considérées. Pour que l'algorithme retourne un résultat correct, E doit être choisi de sorte que le problème associé soit inversible.

Les données en sortie de l'algorithme sont :

- $sol_{P,B}(E) \subset E$ l'ensemble de toutes les configurations de E qui sont *autorisées* par rapport à P et B. Celles-ci sont notées a, le fait qu'elles soient autorisées signifie qu'elles vérifient : $\bigcup B(a) \cap P = \emptyset$.

Nous détaillons quelques-unes des étapes importantes de l'algorithme générique pour en faciliter la compréhension.

1. **Initialisation** L'étape initiale dépend fortement du problème de placement traité. Néanmoins, comme il s'agit de déterminer les configurations qui empêchent un et un seul disque d'être vide, la rotation

n'a pas d'influence et le calcul est alors facilité. Puisque l'ensemble A prend lui-même la forme d'une zone interdite, son stockage ne nécessite pas de structure de données dédiée, ni de calculs complexes.

2. **Étape itérative #1** Le calcul d'une zone interdite élémentaire $\delta_{i,p_j}(E)$ constitue la clé du procédé de calcul. Il est exécuté à chaque itération de la boucle intérieure, donc pour tout couple : disque $b_i \in B$, point $p_j \in P$ avec $2 \leq i \leq m$.
3. **Étape itérative #2** La seconde étape itérative est importante dans la mesure où elle réunit, collecte les contraintes de placement générées une à une par les disques b_i. Elle consiste à appliquer la fonction-centre réciproque sur une zone interdite élémentaire. Elle enregistre le résultat en l'ajoutant à la liste $tabImgZI$.
4. **Étape finale** On calcule finalement le complément ensembliste de l'union de A et de tous les ensembles listés par $tabImgZI$, par rapport à E. En pratique, elle est relativement simple. Elle consiste par exemple à inverser le signe d'inégalités. Concernant l'occupation mémoire de ce résultat, elle peut être améliorée comme expliqué dans la section 6.2.4.

Pour justifier notre affirmation sur le caractère générique de cet algorithme, nous avons prouvé sa validité.

Validité de l'algorithme générique

Dans cette partie nous donnons une preuve de validité de l'algorithme générique décrit plus tôt. On parle de *preuve de validité* lorsqu'il s'agit de démontrer que les données fournies en sortie d'un algorithme sont identiques à la définition mathématique de la solution. Dans notre contexte, la solution est l'ensemble $sol_{P,B}(E)$ de toutes les configurations autorisées (la notion de configuration autorisée a été donnée par la définition 5.2.1).

Pour établir la validité, on passe par deux propriétés préliminaires. Pour chacune d'elles, on considère que P est un nuage de points, B un ensemble de disques standard composé de m disques, et E un *espace de configurations* qui fait que le problème de placement correspondant est un problème inversible. De plus, nous notons $E' \subset E$ l'ensemble des configurations qui engendrent des copies dont le premier disque est vide *i.e.* $b_1(a) \cap P = \emptyset$.

Propriété 6.2.7 (Détection de non appartenance au disque). *Soit a une configuration de E'. Dire qu'un point $p_j \in P$ n'appartient pas au disque $b_i(a)$ de la copie $\cup B(a)$, équivaut à dire que le centre $c_i(a)$ du i-ème disque n'appartient pas à la zone interdite élémentaire $\delta_{i,p_j}(E)$. Formellement :*

$$p_j \notin b_i(a) \Leftrightarrow c_i(a) \notin \delta_{i,p_j}(E)$$

Démonstration. Rappelons la définition d'une orbite :

$$orb_i(E) = \left\{ y \in \mathbb{R}^2 \mid \exists a \in E; y = c_i(a) \text{ et } b_1(a) \cap P = \emptyset \right\}$$

Reprenons aussi la définition 6.2.5 de zone interdite élémentaire :

$$\delta_{i,p_j}(E) := \left\{ z \in orb_i(E) \mid p_j \in b_i(c_i^{-1}(z)) \right\}$$

Ces définitions sont valables pour tout i entre 2 et m, l'entier m étant le nombre de disques composant l'EDS B. De la définition de zone interdite élémentaire, on déduit l'équivalence suivante :

$$\begin{aligned} c_i(a) \in \delta_{i,p_j}(E) &\Leftrightarrow c_i(a) \in orb_i(E) \text{ et } p_j \in b_i(a) \\ &\Leftrightarrow b_1(a) \cap P = \emptyset \text{ et } p_j \in b_i(a) \end{aligned}$$

En appliquant l'hypothèse que $a \in E'$, le terme avant l'opérateur logique « et » est toujours vrai, donc :

$$c_i(a) \in \delta_{i,p_j}(E) \Leftrightarrow p_j \in b_i(a)$$

Par négation des membres de cette équivalence, on conclut. □

On vient d'exposer une relation qui lie zone interdite élémentaire et appartenance d'un point $p_j \in P$ à un disque. Un résultat analogue peut être établi pour les zones interdites. Comme expliqué ci-dessous, cette nouvelle propriété lie la notion de zone interdite et le caractère vide d'un disque par rapport à P. Elle concerne cette fois le nuage de points P dans son intégralité.

Propriété 6.2.8 (Détection de disque vide). *Soit a une configuration de E'. Dire que le i-ème disque d'une copie $\cup B(a)$ est vide, équivaut à dire que le centre $c_i(a)$ du i-ème disque n'appartient pas à la zone interdite $\Delta_i(E)$. Formellement :*

$$b_i(a) \cap P = \emptyset \Leftrightarrow c_i(a) \notin \Delta_i(E)$$

Démonstration. Supposons $c_i(a) \notin \Delta_i(E)$ et voyons les conséquences. Du fait de la propriété 6.2.6 de décomposition (propriété qui dit, rappelons-le qu'une zone interdite peut être définie comme l'union de plusieurs zones interdites élémentaires), on déduit que :

$$\begin{aligned} c_i(a) \notin \Delta_i(E) &\Leftrightarrow c_i(a) \notin \bigcup_{p_j \in P} \delta_{i,p_j}(E) \\ &\Leftrightarrow \forall p_j \in P, c_i(a) \notin \delta_{i,p_j}(E) \end{aligned}$$

d'après la propriété 6.2.7 de détection de non appartenance au disque vue précédemment, on a :

$$\Leftrightarrow \forall p_j \in P, p_j \notin b_i(a)$$

finalement :

$$c_i(a) \notin \Delta_i(E) \Leftrightarrow b_i(a) \cap P = \emptyset$$

□

La propriété que nous venons de présenter : détection de disque vide permet d'établir que le résultat calculé par l'algorithme 6 correspond bien à l'ensemble de solutions $sol_{P,B}(E)$. Elle constitue un outil pour prouver la validité de l'algorithme, ce que l'on présente ci-dessous.

Théorème 6.2.9 (validité de l'algorithme générique). *L'ensemble E' dénote les configurations de E tel que le premier disque $b_1(a)$ soit vide. L'espace de configuration E est associé à un problème inversible (la fonction-centre c_i restreinte à E est bijective). Dans ce contexte, l'ensemble $sol_{P,B}$ (E) des configurations autorisées est égal à E' privé de l'union des images respectives de chaque zone interdite, pour i allant de 2 à m. Ces images sont obtenues par application des fonctions-centre inverses. Formellement :*

$$sol_{P,B}(E) = E' \setminus \bigcup_{2 \leq i \leq m} c_i^{-1}(\Delta_i(E))$$

Démonstration. Pour simplifier les notations de la présente démonstration, on introduit le symbole

$$F = \left(E' \setminus \bigcup_{2 \leq i \leq m} c_i^{-1}(\Delta_i(E)) \right)$$

Démontrons l'égalité énoncée par le théorème, en interprétant l'expression de F ; le membre de droite. Supposons qu'une configuration a quelconque de E appartienne à F. Qu'est-ce que cela signifie ? Du fait de l'opération de différence ensembliste, il vient que :

$$\begin{aligned} a \in F &\Leftrightarrow a \in E' \text{ et } a \notin \bigcup_{2 \leq i \leq m} c_i^{-1}(\Delta_i(E)) \\ &\Leftrightarrow a \in E' \text{ et } \forall 2 \leq i \leq m, a \notin c_i^{-1}(\Delta_i(E)) \end{aligned}$$

on applique la fonction-centre, qui étant bijective dans le cas des problèmes inversibles, ne modifie pas la signification du dernier terme de la proposition. Cela donne :

$$\Leftrightarrow a \in E' \text{ et } \forall 2 \leq i \leq m, c_i(a) \notin \Delta_i(E)$$

d'après la propriété 6.2.8 de détection disque vide, on a :

$$\Leftrightarrow a \in E' \text{ et } \forall 2 \leq i \leq m, b_i(a) \cap P = \emptyset$$

on remplace l'appartenance à E' par ce que donne sa définition

$$\Leftrightarrow b_1(a) \cap P = \emptyset \text{ et } \forall 2 \leq i \leq m, b_i(a) \cap P = \emptyset$$

comme tout disque composant la copie est vide de P, alors la copie elle même est vide. Ainsi,

$$a \in F \Leftrightarrow \cup B(a) \cap P = \emptyset$$

Finalement, par définition de l'ensemble des solutions, $sol_{P,B}(E) = F$. □

Le théorème ci-dessus démontre que le procédé de calcul que nous avons proposé permet de résoudre tout PP inversible (en particulier, le placement par rotation). Donc l'algorithme 6 est valide *et* générique.

Dans la partie suivante, on vérifiera qu'il est aussi utilisable en pratique, en analysant sa complexité algorithmique.

Complexité algorithmique en temps

Avant d'implémenter l'algorithme 6 vu précédemment, nous avons étudié sa complexité en temps. Dans cette partie nous présentons le résultat de cette étude. Pour déterminer la *complexité algorithmique*, la remarque qui suit facilite grandement l'analyse.

Remarque. En observant les étapes de l'algorithme générique, on constate qu'à chaque itération de boucle, on opère uniquement via les notions de zones interdites élémentaires et de fonction-centre réciproque. Le calcul des zones interdites et ces opérations associées (calcul de l'image par la fonction-centre réciproque) est réalisé en temps constant, son temps d'exécution ne dépend ni de la taille de P ni de celle de B.

Théorème 6.2.10 (Complexité en temps de l'algorithme générique)**.** *Le nombre d'étapes de calcul est proportionnel au produit $n \times m$ du nombre de points dans P multiplié par le nombre de disques composant l'EDS à placer B. Étant donné P un nuage de n points et B un ensemble standard de m disques, l'algorithme 6 s'exécute donc en $O(nm)$ dans le pire des cas.*

Démonstration. Suivant la remarque donnée ci-dessus, le temps d'exécution de l'algorithme générique pour un PP inversible donné sera proportionnel au nombre de zones interdites à traiter. Par définition, toute zone interdite élémentaire est associée à un couple (i, p_j), avec i l'indice représentant le i-ème disque (ou la i-ème orbite, c'est équivalent) et p_j un élément de P. Puisque $n = card(P)$ et $m = card(B)$, le nombre de zones interdites élémentaires distinctes est égal à $n \times m$. □

L'étude de complexité que l'on vient de décrire montre que l'algorithme générique proposé s'exécute en temps raisonnable et est donc utilisable en pratique, sur machine. La codage sur machine de zones interdites élémentaires varie nécessairement d'un problème inversible à l'autre, toutefois on a vu que les opérations associées (calcul, application de fonction-centre ou fonction-centre réciproque) s'exécutent en temps constant. Dans la sous-section qui suit, nous discutons l'occupation en mémoire des résultats et de possibles post-traitements pouvant s'avérer utiles.

6.2.4 Compactage des résultats

L'algorithme générique de résolution d'un PP inversible a été présenté plus tôt.

On a vu que ce dernier opère sur des *zones interdites* pour produire le résultat : l'ensemble des configurations valides.

Puisque le nombre de zones interdites est connu à l'avance (on rappelle qu'il s'agit du produit $m \times n$ où m représente le nombre de disques constituant l'EDS reçu en entrée, et n représente le nombre de points dans l'ensemble P), le résultat obtenu possède un nombre identique de « constituants » (chaque constituant étant l'image d'une zone interdite par la fonction-centre inverse).

Propriété 6.2.11. *Le stockage sur ordinateur de l'ensemble-solution $sol_{P,B}(E)$ exigera un espace mémoire proportionnel au nombre de constituants ; « constituant » signifiant image d'une zone interdite par application de la fonction centre-*

inverse c_i^{-1}. Ceci est valable quelque soit le PP *considéré, comme toutes les opérations de calcul découlent de la notion générique de zone interdite.*

Selon le problème de placement et le jeu de données P, B considérés, le coût de stockage des résultats peut être réduit *a posteriori*. En éliminant l'information redondante, c'est-à-dire les zones interdites qui se recouvrent les unes les autres, il est possible de compacter la représentation de l'ensemble-solution. Cela peut s'avérer utile voire même nécessaire selon l'usage que l'on fait des solutions, selon le domaine applicatif.

Ainsi les résultats retournés par l'algorithme peuvent donc être « compactés » dans de nombreux cas de figure. Ils le peuvent à la condition qu'une fois que les zones interdites ont été calculées et ramenées à des sous-ensembles de l'espace de configuration, plusieurs de ces sous-ensembles aient une intersection non vide ou bien qu'un sous-ensemble soit inclus dans un autre.

Voyons par exemple le placement utilisant la rotation uniquement, les zones interdites étant dans ce cas précis des arcs de cercle, la réduction du coût de stockage peut être simplement réalisée en faisant l'union des arcs qui sont non disjoints. De manière globale, les configurations qui correspondent à n'importe quel arc interdit prennent la forme : $a = (u_k, \lambda_k, \theta), \theta \in [x_{inf}, x_{sup}]$. Les intervalles $[x_{inf}, x_{sup}] \subset [0, 2\pi[$ pouvant se chevaucher, il devient inutile de stocker certains d'entre eux. On peut alors leur appliquer un procédé de fusion pour « compacter » ainsi la représentation des solutions.

Concrètement, soit $I_1, I_2, \ldots, I_k$ des intervalles de $[0, 2\pi[$. Chaque intervalle peut s'écrire $I_l = [x_{linf}, x_{lsup}]$ Un procédé de fusion possible est le suivant :

1. trier les bornes x_{linf}, x_{lsup} indifféremment, et ce pour tout $1 \leq l \leq k$
2. déclarer une variable-compteur notée `cpt` et l'initialiser à 0. Déclarer une variable temporaire x_{tmp}
3. Parcourir la suite $(x_1, x_2, \ldots, x_{2k})$ obtenue après le tri, tout en mettant à jour la variable-compteur comme suit : Lorsque x_l est une borne inférieure alors on incrémente la variable `cpt`, sinon on la décrémente.
4. À chaque fois que `cpt` passe de 0 à 1 on enregistre x_l dans x_{tmp}
5. À chaque fois que `cpt` passe de 1 à 0 on crée un nouvel intervalle $[x_{tmp}, x_l]$

Ce procédé peut même être accompli en temps $O(nm)$ au plus si les arcs interdits ont été insérés dans un ordre trié au fur et à mesure du calcul. Expérimentalement, nous avons constaté que pour ce PP, le coût de stockage de l'ensemble-solution peut être fortement réduit. La réduction de l'espace de stockage peut aller jusqu'à 60%. Cette dernière quantité dépend fortement du jeu de données utilisé.

Malgré ce haut niveau d'abstraction, l'algorithme est utilisable en pratique. Pour preuve, après présentation de la théorie nous appliquons la méthode à trois problèmes de placement. Pour chacun d'eux, la mise en œuvre et les résultats obtenus sont présentés dans la section suivante.

6.3 Exemples d'application de la méthode

Plusieurs problèmes de placement d'union de disques ont été mentionnés précédemment, et nous avons proposé une méthode générique pour les résoudre. Après avoir conçu cette méthode, pour vérifier qu'elle est utilisable en pratique, nous l'avons appliquée à différents PP. A titre d'exemple, dans cette section, nous montrons comment procéder pour les trois problèmes que sont :

- le placement par homothétie uniquement ;
- le placement par homothétie et rotation combinée ;
- le placement par translation uniquement.

À chaque fois, nous partons de la définition mathématique et abstraite, et une fois l'ensemble E des configurations considérées choisi, à l'aide de déductions nous précisons ces notions jusqu'à arriver à des objets géométriques simples (disques, segments, *etc.*)

Plus schématiquement, nous procédons en trois étapes :

1. vérification du caractère inversible du problème ;
2. calcul des configurations produisant des copies ayant leur premier disque vide ;
3. conséquences de la définition d'orbite générale et de zone interdite.

Ce procédé sera appliqué à différents PP dans chacune des trois prochaines sous-sections. De cette manière nous sommes parvenus à la représentation et au calcul effectif par ordinateur des ensembles-solution. Dans tout ce qui suit B dénote un EDS et P un nuage de points.

6.3.1 Placement par homothétie uniquement

Le placement par homothétie est un problème inversible. Lorsque la fonction c_i est restreinte au domaine $E_{H.}$, la fonction-centre réciproque c_i^{-1} existe. Elle est définie comme suit, pour tout $2 \leq i \leq m$:

$$\begin{array}{rccl} c_i^{-1}: & \mathbb{R}^2 & \to & E_{H.} \\ & y & \mapsto & \left(u_k, \frac{\|y-u_k\|}{\|c_i\|}, \theta_k\right) \end{array}$$

En effet, vérifions que $c_i^{-1}(c_i(a)) = a$. Soit a une configuration de $E_{H.}$, donc a est de la forme : (x_k, λ, θ_k), avec d'une part $\lambda \in \mathbb{R}^+$ variable, d'autre part x_k, θ_k constants. Il vient que :

$$\begin{array}{rcl} c_i^{-1}(c_i(a)) & = & c_i^{-1}\left(x_k + \lambda \mathbf{M}_{\theta_k} c_i\right) \\ & = & \left(x_k, \frac{\left\|x_k + \lambda \mathbf{M}_{\theta_k} c_i - x_k\right\|}{\|c_i\|}, \theta_k\right) \\ & = & \left(x_k, \frac{\left\|\lambda \mathbf{M}_{\theta_k} c_i\right\|}{\|c_i\|}, \theta_k\right) = (x_k, \lambda, \theta_k) = a \end{array}$$

Configurations produisant des copies ayant le premier disque vide

Il s'agit de calculer le sous-ensemble $E' \subset E$ des configurations qui produisent une copie $\cup B(a)$ ayant son premier disque vide. Formellement :

$$E' = \{a \in E \mid b_1(a) \cap P = \emptyset\}$$

Dans le cas de l'homothétie, le sous-ensemble noté $E'_{H.}$ est égal à :

$$E'_{H.} = \{u_k\} \times [0, min_{p_l \in P} (\|p_l - u_k\|)] \times \{\theta_k\}$$

En effet, comme :

$$b_1(a) \cap P = \emptyset \Leftrightarrow \forall p_i \in P, p_i \notin b_1(a)$$

Dans notre cas particulier, en notant $a = (u_k, \lambda, \theta_k)$, il vient que :

$$\begin{aligned} b_1(a) \cap P = \emptyset &\Leftrightarrow \forall p_i \in P, p_i \notin b_1(a) \\ &\Leftrightarrow \forall p_i \in P, \|p_i - c_1(a)\| \geq r_1(a) \\ &\Leftrightarrow \forall p_i \in P, \|p_i - u_k\| \geq \lambda \\ &\Leftrightarrow min_{p_l \in P} (\|p_l - u_k\|) \geq \lambda \end{aligned}$$

Orbite dans le cas du placement par homothétie

La définition 6.2.2 d'orbite générale a été donnée précédemment, pour tout $2 \leq i \leq m$. Dans le cas particulier du placement par homothétie, les *orbites* consistent en des segments de droite comme on le montre ci-dessous. On sait que :

$$orb_i (E_{H.}) = c_i(E'_{H.})$$

En utilisant l'expression de $E'_{H.}$ vue plus haut, il vient que :

$$c_i \left(E'_{H.}\right) = \left\{z \in \mathbb{R}^2 \mid z = x_k + \lambda \mathbf{M}_{\theta_k} c_i \text{ où } \lambda \in [0, min_{p_l \in P} (\|p_l - u_k\|)]\right\}$$

Par conséquent, une orbite $orb_i (E_{H.})$ consiste en un segment de droite $[A, B]$ d'extrémités A, B où A et B sont des points définis tels que :

$$\left\{ \begin{array}{ll} A & = u_k \\ B & = u_k + min_{p_l \in P} (\|p_l - u_k\|) M_{\theta_k} c_i \end{array} \right.$$

Zone interdite dans le cas du placement par homothétie

Reprenons la définition 6.2.4 de zone interdite élémentaire donnée précédemment :

$$\delta_{i,p_i}(E) := \left\{x \in orb_i(E) \mid p_j \in b_i(c_i^{-1}(x))\right\}$$

Cette définition s'applique à tout problème de placement inversible quel qu'il soit. En posant que $E = E_{H.}$, en particulier il vient que $x \in \delta_{i,p_j}(E_{H.})$ est équivalent à :

$$x = (u_k + \lambda \mathbf{M}_{\theta_k} c_i) \text{ avec } 0 \leq \lambda \leq min_{p_l \in P} (\|p_l - u_k\|) \text{ et } \left\|x - p_j\right\| < \lambda r_i$$

La première ligne traduit l'appartenance de x à l'orbite *i.e.* $x \in orb_i(E)$, la seconde ligne traduit l'appartenance de p_j au i-ème disque d'une copie

particulière *i.e.* $p_j \in b_i(c_i^{-1}(x))$. En injectant l'information de la première ligne dans la seconde, il vient que :

$$\begin{aligned} \left\|x_k + \lambda \mathbf{M}_{\theta_k} c_i - p_j\right\| &< \lambda r_i \\ \Leftrightarrow \left\|x_k + \lambda \mathbf{M}_{\theta_k} c_i - p_j\right\|^2 &< \lambda^2 r_i^2 \\ \Leftrightarrow \left\|x_k - p_j\right\|^2 + 2\langle x_k - p_j; \lambda \mathbf{M}_{\theta_k} c_i\rangle + \left\|\lambda \mathbf{M}_{\theta_k} c_j\right\|^2 - \lambda^2 r_i^2 &< 0 \\ \Leftrightarrow \lambda^2 \left(\left\|c_j\right\|^2 - r_j^2\right) + 2\lambda\langle x_k - p_j; \mathbf{M}_{\theta_k} c_i\rangle + \left\|x_k - p_j\right\|^2 &< 0 \end{aligned}$$

Dans le cas du placement par homothétie, la zone interdite élémentaire pour un couple $b_i \in B, p_j \in P$ peut alors être décrite comme l'ensemble des points membres de l'orbite $orb_i(E_{H.})$ et vérifiant l'inéquation suivante :

$$\lambda^2 \left(\left\|c_j\right\|^2 - r_j^2\right) + 2\lambda\langle u_k - p_i; \mathbf{M}_{\theta_k} c_j\rangle + \left\|u_k - p_i\right\|^2 < 0$$

Dans tous les cas, une zone interdite élémentaire consiste soit en l'ensemble vide, soit en un segment de droite, soit en une demi-droite. La première extremité du-dit segment est soit u_k soit une solution à la précédente équation. La seconde extrémité du-dit segment est soit $min_{p_l \in P}\left(\left\|p_l - u_k\right\|\right)$ soit une solution à la précédente équation.

L'implémentation fournie en annexe A.5 fournit les détails du traitement cas par cas. Un exemple de résultats obtenus est représenté figure 6.7.

6.3.2 Placement par homothétie et rotation combinées

Le placement par homothétie-rotation combinée est un problème inversible. Lorsque la fonction c_i est restreinte au domaine $E_{R.H.}$, la fonction-centre réciproque c_i^{-1} existe. Elle est définie ainsi, pour tout $2 \leq i \leq m$:

$$\begin{array}{rccl} c_i^{-1}: & \mathbb{R}^2 & \to & E_{R.H.} \\ & y & \mapsto & \left(x_k, \frac{\|y - x_k\|}{\|c_i\|}, \angle\left(\overrightarrow{x_k X}, \overrightarrow{x_k y}\right) - \angle\left(\overrightarrow{\mathcal{O}X}, \overrightarrow{\mathcal{O}c_i}\right)\right) \end{array}$$

En effet, vérifions que $c_i^{-1}(c_i(a)) = a$. Soit a une configuration de $E_{R.H.}$, donc a est de la forme : (x_k, λ, θ), avec d'une part $\lambda \in \mathbb{R}^+$ et $\theta \in [0, 2\pi[$ variables, d'autre part x_k constant. Il vient que :

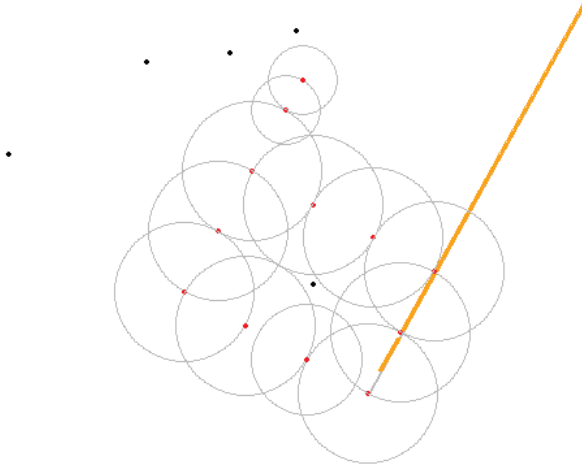
$$\begin{array}{rcl} c_i^{-1}(c_i(a)) & = & c_i^{-1}\left(x_k + \lambda \mathbf{M}_\theta c_i\right) \\ & = & \left(x_k, \frac{\|x_k + \lambda \mathbf{M}_\theta c_i - x_k\|}{\|c_i\|}, \angle\left(\overrightarrow{x_k X}, \overrightarrow{x_k\left(x_k + \lambda \mathbf{M}_\theta c_i\right)}\right) - \angle\left(\overrightarrow{\mathcal{O}X}, \overrightarrow{\mathcal{O}c_i}\right)\right) \\ & = & \left(x_k, \frac{\|\lambda c_i\|}{\|c_i\|}, \angle\left(\overrightarrow{\mathcal{O}X}, \overrightarrow{\mathcal{O}\left(\lambda \mathbf{M}_\theta c_i\right)}\right) - \angle\left(\overrightarrow{\mathcal{O}X}, \overrightarrow{\mathcal{O}c_i}\right)\right) \\ & = & \left(x_k, \lambda, \theta + \angle\left(\overrightarrow{\mathcal{O}X}, \overrightarrow{\mathcal{O}c_i}\right) - \angle\left(\overrightarrow{\mathcal{O}X}, \overrightarrow{\mathcal{O}c_i}\right)\right) \\ & = & (x_k, \lambda, \theta) = a \end{array}$$

Le PP par homothétie-rotation combinée est bien inversible, on peut donc le résoudre avec l'algorithme générique 6.

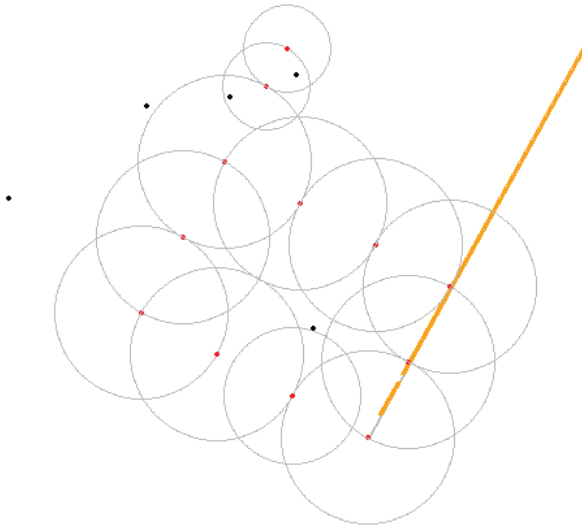
Configurations produisant des copies ayant le premier disque vide

Calculons le sous-ensemble $E'_{R.H.}$ des configurations qui produisent une forme $B(a)$ ayant son premier disque vide. Formellement :

$$E'_{R.H.} = \{a \in E_{R.H.} \mid b_1(a) \cap P = \emptyset\}$$



(a)



(b)

L'ensemble des solutions obtenu décrit toutes les possibilités de placer l'union de disques $\cup B$, en utilisant l'homothétie uniquement, en évitant le nuage de points. Dans le contexte du placement par homothétie, les orbites sont des demi-droites (représentées en gris) et les arcs interdits des segments ou des demi-droites (représentés en trait épais orange).

FIGURE 6.7 – *Solution fournie par l'algorithme 6 pour un* PP *par homothétie*

Nous suivons exactement le même raisonnement que pour le problème de placement par homothétie, qui a été présenté plus haut. Il vient que :

$$E'_{R.H.} = \{u_k\} \times [0, min_{p_l \in P}(\|p_l - u_k\|)] \times \{\theta\} \text{ où } \lambda, \theta \text{ varient librement}$$

Cela s'explique du fait que :

$$\begin{aligned} \forall a \in E_{R.H.}, b_1(a) \cap P = \emptyset &\Leftrightarrow \forall p_i \in P, p_i \notin b_1(a) \\ &\Leftrightarrow \forall p_i \in P, \|p_i - c_1(a)\| \geq r_1(a) \\ &\Leftrightarrow \forall p_i \in P, \|p_i - u_k\| \geq \lambda \\ &\Leftrightarrow min_{p_l \in P}(\|p_l - u_k\|) \geq \lambda \end{aligned}$$

Orbite dans le cas du placement par homothétie-rotation combinée

La définition 6.2.2 d'orbite générale a été donnée précédemment, pour tout $2 \leq i \leq m$. Dans le cas particulier du placement par homothétie-rotation combinée, les *orbites* consistent en des disques fermés centrés en x_k. En effet, nous savons que :

$$orb_i(E_{R.H.}) = c_i\left(E'_{R.H.}\right)$$

D'après le résultat précédent (configurations vérifiant la condition initiale) on sait que λ doit être borné par $min_{p_l \in P}(\|p_l - u_k\|)$ pour que le premier disque $b_1(a)$ soit vide. Donc pour que la configuration associée a appartienne à $E'_{R.H.}$. Ainsi pour $2 \leq i \leq m$ il vient que :

$$\begin{aligned} orb_i(E_{R.H.}) &= \left\{x \in \mathbb{R}^2 \mid x = u_k + \lambda \mathbf{M}_\theta c_i, \lambda \leq min_{p_l \in P}(\|p_l - u_k\|)\right\} \\ &= \overline{b}\left(u_k, min_{p_l \in P}(\|p_l - u_k\|)\right) \end{aligned}$$

Où $\overline{b}$(c,r) dénote un disque fermé et $A \subset \mathbb{R}^2$ une région quelconque du plan, $\overline{A}$ représente sa fermeture.

Zone interdite dans le cas du placement par homothétie-rotation combinée

Concernant la définition de zone interdite, on va voir que la zone interdite élémentaire n'est composée que de disques et de demi-plans.

En effet, dans notre cas particulier et en réutilisant le résultat précédent concernant l'orbite, il vient que :

$$\begin{aligned} \delta_{i,p_j}(E_{R.H.}) &:= \{x \in orb_i(E_{R.H.}) \mid p_j \in b_i(c_i^{-1}(x))\} \\ &= orb_i(E_{R.H.}) \cap A \\ &= \overline{b}\left(x_k, min_{p_l \in P}(\|p_l - u_k\|)\right) \cap A \end{aligned}$$

Où A est un ensemble égal à :

$$\begin{aligned} A &= \{x \in \mathbb{R}^2 \mid \left\|p_j - c_j(c_i^{-1}(x)\right\| < r_i(c_i^{-1}(x))\} \\ &= \{x \in \mathbb{R}^2 \mid \left\|p_j - x\right\| < \frac{\|u_k - x\|}{\|c_i\|} r_i\} \end{aligned} \tag{6.2}$$

Propriété 6.3.1 (interprétation de la zone intedite, cas homothétie-rotation)**.** *Toute zone interdite élémentaire $\delta_{i,p_j}(E_{R.H.})$, dans le cas du placement par homothétie-rotation combinées n'est composée que de disques et de demi-plans. En effet :*

- *si $\|c_i\| > r_i$ alors A est égal à un disque ouvert $b(C,R)$ où : $C = \left(\frac{\|c_i\|^2 p_x - r_i^2 v_x}{\|c_i\|^2 - r_i^2}, \frac{\|c_i\|^2 p_y - r_i^2 v_y}{\|c_i\|^2 - r_i^2}\right)$ et $R = \|p - v\|^2 \left(\frac{\|c_i\| r_i}{\|c_i\|^2 - r_i^2}\right)$*
- *si $\|c_i\| = r_i$ alors A est égal à un demi-plan de vecteur directeur $\overrightarrow{x_k p}$ et passant par le point O situé à égale distance de p et x_k tel que : $O = (p + x_k)/2$*
- *si $\|c_i\| < r_i$ alors A est égal au complément du disque fermé $\overline{b}(C,R)$ où C et R sont définis comme précédemment.*

La justification de cette interprétation est donnée en annexe A.6.

Une représentation graphique pertinente des solutions est possible en calculant $c_2(sol_{P,B}(E_{R.H.}))$. Grâce à la bijectivité de la fonction-centre c_i, on ne perd pas d'informations en réalisant une telle projection vers le plan. Dans le cas du placement par homothétie-rotation, nous avons :

$$c_2(sol_{P,B}) = \left\{z \in \mathbb{R}^2 \mid \exists a \in E_{R.H.}, z = c_2(a) \text{ et } \cup B(a) \cap P = \emptyset\right\}$$

Pour calculer cet ensemble, nous avons utilisé les orbites, zones interdites, et fonctions-centre inverses introduites précédemment. Nous donnons quelques exemples de solutions au PP homothétie-rotation combinées, calculées par notre algorithme. Elles sont représentées sur les figures 6.8, 6.9 et 6.10.

6.3.3 Placement par translation uniquement

Le placement par translation est un problème inversible. Lorsque la fonction c_i est restreinte au domaine $E_{T.}$, La fonction-centre réciproque c_i^{-1} existe. Elle est définie comme suit, pour tout $2 \leq i \leq m$:

$$\begin{array}{rccl} c_i^{-1}: & \mathbb{R}^2 & \rightarrow & E_{T.} \\ & y & \mapsto & (y - \lambda_k \mathbf{M}_{\theta_k} c_i, \lambda_k, \theta_k) \end{array}$$

En effet, soient B un EDS et a une configuration de $E_{T.}$, par conséquent a peut s'écrire sous la forme : (x, λ_k, θ_k), où λ_k, θ_k sont fixés. Vérifions que $c_i^{-1}(c_i(a)) = a$:

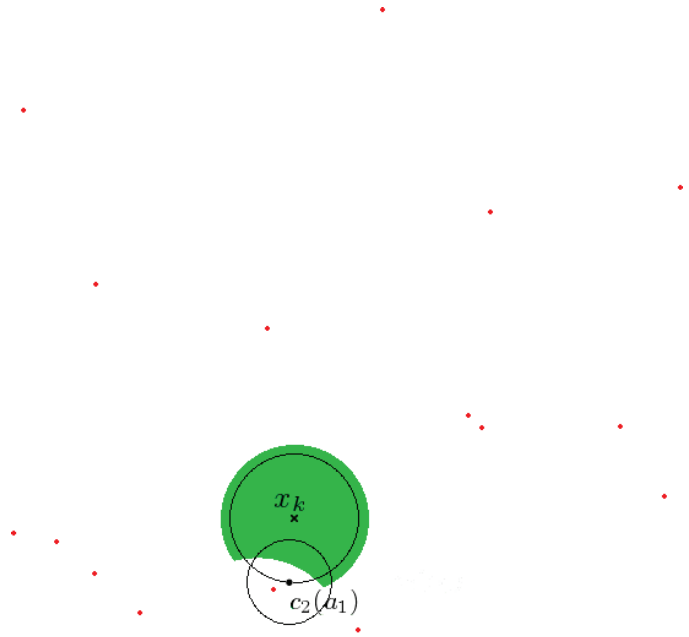
$$\begin{array}{rcl} c_i^{-1}(c_i(a)) & = & c_i^{-1}\left(x + \lambda_k \mathbf{M}_{\theta_k} c_i\right) \\ & = & (x + \lambda_k \mathbf{M}_{\theta_k} c_i - \lambda_k \mathbf{M}_{\theta_k} c_i, \lambda_k, \theta_k) \\ & = & (x, \lambda_k, \theta_k) = a \end{array}$$

Le PP par translation est bien inversible, notre méthode de résolution générique peut lui être appliqué.

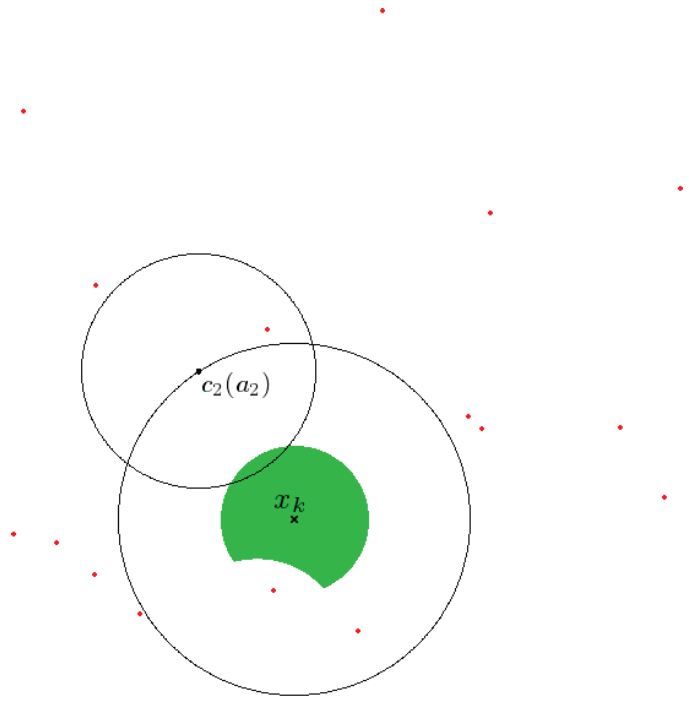
Configurations produisant des copies ayant le premier disque vide

Il s'agit de calculer le sous-ensemble $E' \subset E$ des configurations qui produisent une copie $\cup B(a)$ ayant son premier disque vide. Formellement :

$$E' = \{a \in E \mid b_1(a) \cap P = \emptyset\}$$



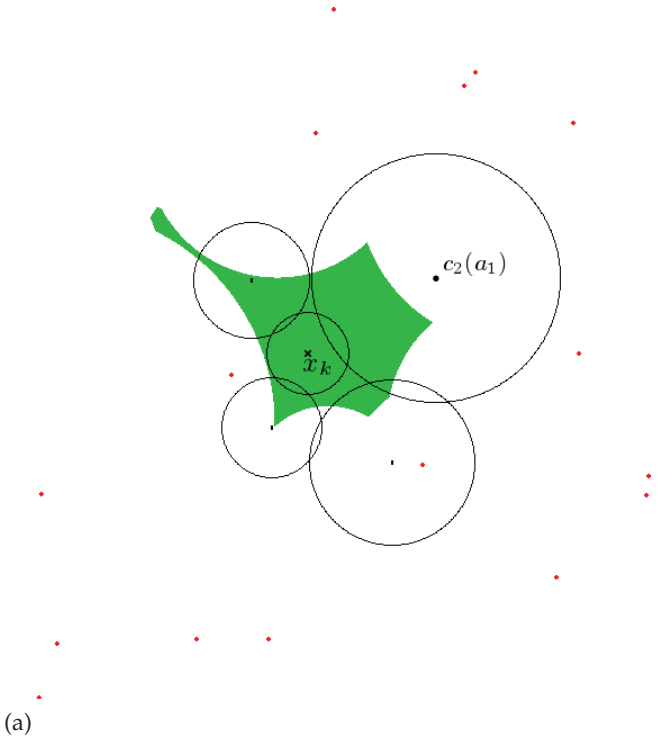
(a)



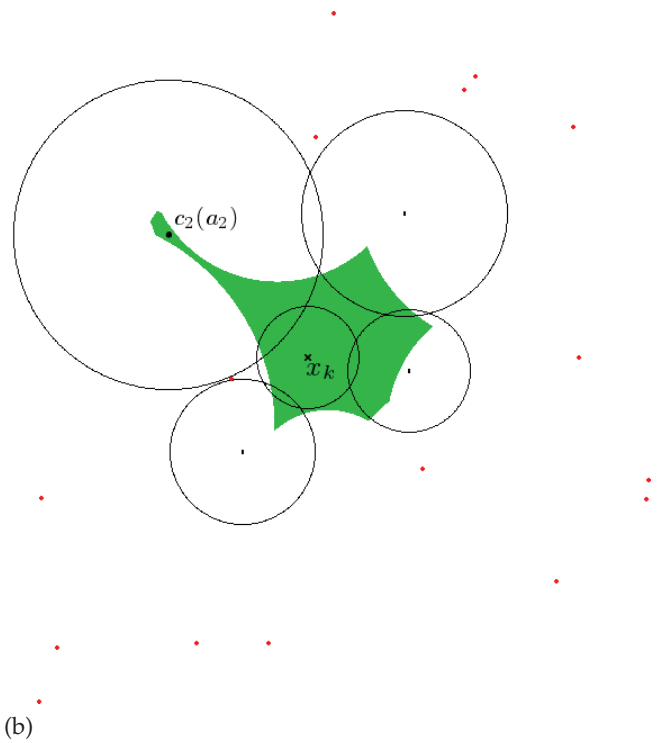
(b)

Notre procédé de calcul détermine les possibilités de placement de l'union de disques $\cup B$, en utilisant l'homothétie et la rotation simultanément, de sorte à éviter le nuage de points. La région représentée en vert représente toutes les positions acceptables pour le centre du deuxième disque

FIGURE 6.8 – *Solution fournie par l'algorithme 6 pour répondre au* PP *par homothétie-rotation combinées (1)*



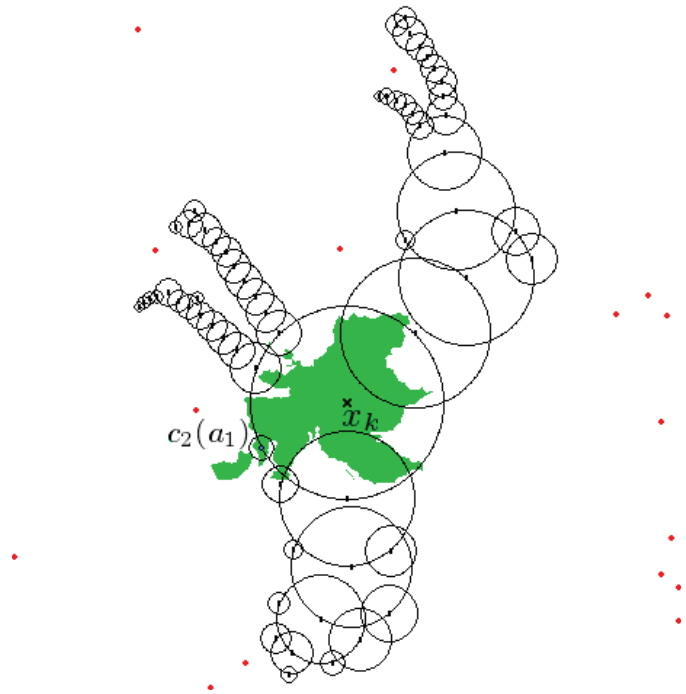
(a)



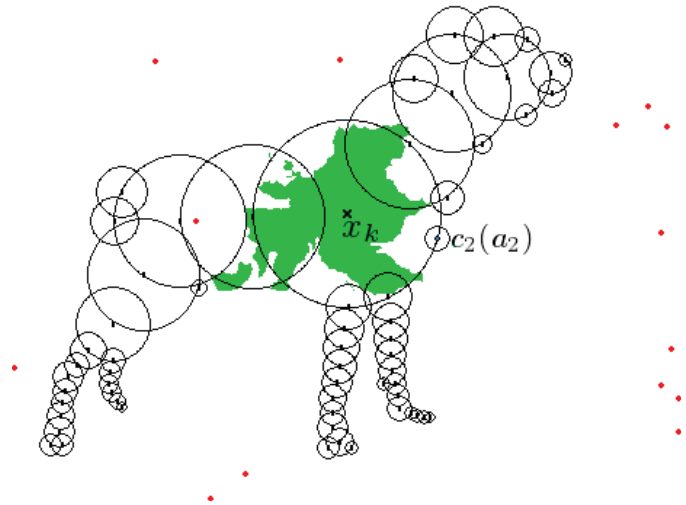
(b)

Notre procédé de calcul détermine les possibilités de placement de l'union de disques $\cup B$, en utilisant l'homothétie et la rotation simultanément, de sorte à éviter le nuage de points. La région représentée en vert représente toutes les positions acceptables pour le centre du deuxième disque

FIGURE 6.9 – *Solution fournie par l'algorithme 6 pour répondre au* PP *par homothétie-rotation combinées (2)*



(a)



(b)

Notre procédé de calcul détermine les possibilités de placement de l'union de disques $\cup B$, en utilisant l'homothétie et la rotation simultanément, de sorte à éviter le nuage de points. La région représentée en vert représente toutes les positions acceptables pour le centre du deuxième disque

FIGURE 6.10 – *Solution fournie par l'algorithme 6 pour répondre au* PP *par homothétie-rotation combinées (3)*

Dans le cas de la translation, cet ensemble noté $E'_{T.}$ est déterminé comme suit.

$$E'_{T.} = \left(\mathbb{R}^2 \setminus \bigcup_{p_j \in P} b(p_j, \lambda_k) \right) \times \{\lambda_k\} \times \{\theta_k\}$$

En effet, comme :

$$b_1(a) \cap P = \emptyset \Leftrightarrow \forall p_j \in P; \left\| p_j - c_1(a) \right\| \geq r_1(a)$$

Dans notre cas particulier il vient que :

$$\begin{aligned} E'_{T.} &= \left\{ (u, \lambda_k, \theta_k) \in E \mid \left\| p_j - u \right\| \geq \lambda_k \right\} \\ &= \left(\mathbb{R}^2 \setminus \bigcup_{p_j \in P} b(p_j, \lambda_k) \right) \times \{\lambda_k\} \times \{\theta_k\} \end{aligned}$$

Orbite dans le cas du placement par translation

La définition 6.2.2 d'orbite valable pour tout problème inversible et pour tout $2 \leq i \leq m$ a été donnée précédemment. Rappelons ci-dessous cette définition, en la précisant pour le cas particulier du placement par translation :

$$\begin{aligned} orb_i c_i(E_{T.}) &= \left\{ y \in \mathbb{R}^2 \mid \exists a \in E_{T.}; y = c_i(a) \text{ et } b_1(a) \cap P = \emptyset \right\} \quad = c_i(E'_{T.}) \\ &= \mathbb{R}^2 \setminus \bigcup_{p_j \in P} b(p_j + \lambda_k \mathbf{M}_{\theta_k} c_i, \lambda_k) \end{aligned}$$

Zone interdite dans le cas du placement par translation

De même, la définition de zone interdite pour tout problème inversible a été donnée dans la définition 6.2.4. Dans le cas particulier de la translation, la zone interdite élémentaire pour un couple $b_i \in B, p_j \in P$ est décrite par l'équivalence :

$$\begin{aligned} x \in \delta_{i,p_j}(E_{T.}) \Leftrightarrow & x \notin \bigcup_{p_j \in P} b(p_j + \lambda_k \mathbf{M}_{\theta_k} c_i, \lambda_k) \\ & \text{et } x \in b\left(p_j - \lambda_k \mathbf{M}_{\theta_k} c_i, \lambda_k r_i\right) \end{aligned}$$

Ainsi, pour le placement par translation la zone interdite élémentaire associée à i, p_j dispose de trois interprétations géométriques différentes selon les constantes $\lambda_k, \theta_k, p_j, c_i$,.. La région $\delta_{i,p_j}(E_{T.})$ peut-être :

- soit le disque ouvert $b\left(p_j - \lambda_k \mathbf{M}_{\theta_k} c_i, \lambda_k r_i\right)$
- soit ce même disque ouvert privé d'un ou plusieurs disques ouverts de la forme $b(p_j, \lambda_k)$, $p_j \in P$
- soit l'ensemble vide

En effet,

$$\delta_{i,p_j}(E_{T.}) = \left\{ x \in orb_i \mid p_j \in b_i\left(c_i^{-1}(x)\right) \right\}$$

Or,

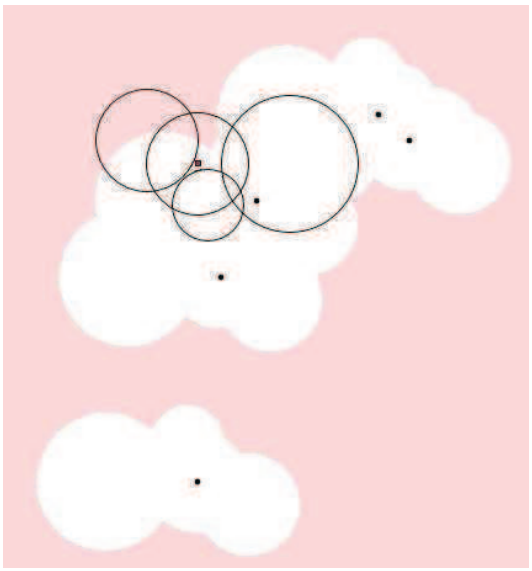
$$\begin{aligned} x \in \delta_{i,p_j}(E_{T.}) \Leftrightarrow & x \in \mathbb{R}^2 \setminus \bigcup_{p_j \in P} b(p_j + \lambda_k \mathbf{M}_{\theta_k} c_i, \lambda_k) \\ & \text{et } \left\| x - \lambda_k \mathbf{M}_{\theta_k} c_i - p_j \right\| < \lambda_k r_i \end{aligned}$$

Ayant explicité ce qu'est la fonction-centre réciproque c_i^{-1}, les orbites et les zones interdites pour le cas particulier du placement par translation, nous avons pu appliquer l'algorithme générique 6 pour résoudre ce problème.

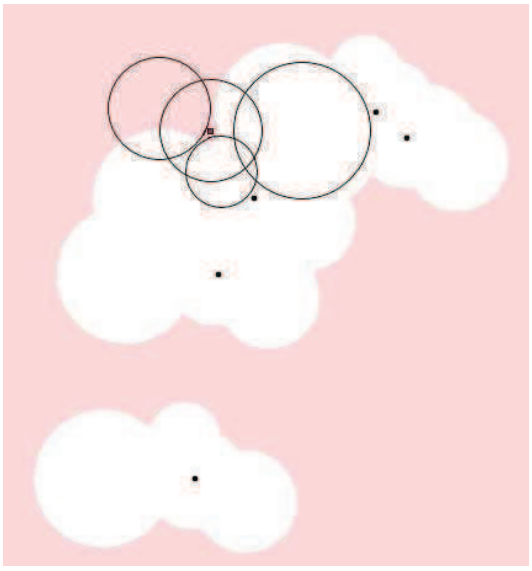
Sur la figure 6.11, nous donnons un exemple de résultats calculés. La zone couleur rose pâle représente le lieu où l'on peut translater le premier centre de la forme, de sorte que cette même forme ne contienne aucun point de P en son intérieur.

Conclusion du chapitre

Ce chapitre a présenté une méthode générique pour la résolution de problèmes de placement d'union de disques, dans un nuage de points. La méthode est générique au sens où elle s'applique et fonctionne indifféremment sur plusieurs PP. À notre connaissance, c'est la première méthode de ce type, les solutions existant dans la littérature étant spécifiques à un PP particulier. Le cadre théorique proposé permet quant à lui d'unifier ces travaux, pour rendre plus commode aussi bien leur comparaison que leur résolution. Nous défendons ce point de vue dans le chapitre suivant qui est la conclusion générale de la thèse.



(a)



(b)

L'ensemble des solutions obtenu décrit toutes les possibilités de placer l'union de disques $\cup B$, en utilisant la translation uniquement, de sorte à ne pas intersecter le nuage de points. Les positions acceptables pour le premier centre sont modélisées par une région représentée ici en rose pâle.

FIGURE 6.11 – *Solution fournie par l'algorithme 6 pour un* PP *par translation*

Conclusion et perspectives

Dans ce mémoire, nous avons présenté deux contributions : une nouvelle généralisation du diagramme de Voronoï et une méthode générique pour le placement de formes géométriques dans un nuage de points.

Premièrement, en généralisant le concept de région de Voronoï, nous avons défini des régions vérifiant un prédicat dit d'union de disques vide. De telles régions, appelées régions de VPUD, sont paramétrées par une union de disques *i.e.* une « forme modèle », en plus d'être associées une à une aux points de P. Plusieurs propriétés des régions de VPUD ont été mises en évidence au chapitre 4. Ces régions ont notamment l'avantage de grouper un ensemble important de copies de l'union de disque, copies dont on sait qu'elles sont vides de tout point. Nous avons alors proposé des méthodes pour représenter ces régions et les calculer sur ordinateur. Les algorithmes en question ont effectivement été prototypés ; nous les avons implémentés par le biais des langages *Python* et *C++*.

Deuxièmement, en proposant une méthode pour placer une forme complexe dans un nuage de points, nous avons introduit un formalisme dédié (articulé autour des notions de copie, de configuration et d'espace de configurations) et des définitions géométriques générales (fonction-centre et fonction-centre réciproque, orbites, zone interdite) qui permettent d'utiliser un seul et même procédé de calcul pour résoudre divers problèmes de placement. En cela, l'algorithme que nous avons décrit au chapitre 6 est générique. Il permet de résoudre tout problème de placement à condition que celui-ci appartienne à la classe des *problèmes inversibles*, comme expliqué dans le théorème 6.2.9. Concrètement, nous avons appliqué cet algorithme pour la résolution du : PP par rotation, PP par homothétie, PP par homothétie-rotation combinées et PP par translation.

Pour le développement des deux thèmes de la thèse, une représentation par union de disques a été employée. Une telle représentation permet, en tolérant une éventuelle approximation, de représenter des formes géométriques complexes et variées par assemblage de disques. Le code-source du PP par homothétie, présenté en annexe A.5, est remarquablement court et efficace. Il témoigne du fait que la représentation par union de disques est simple et pratique.

Le fait qu'une union de disques soit une région décomposable en une multitude « d'éléments de base », que ceux-ci puissent être traités de manière totalement indépendante, laisse entrevoir d'intéressantes perspectives algorithmiques. Nos prototypes logiciels actuels opèrent de manière incrémentale. Dans la continuité de nos travaux, nous aimerions explorer une autre voie : paralléliser les algorithmes de placement pour tirer le plein avantage de la représentation par union de disques, en conjonction avec les architectures matérielles présentement disponibles (processeurs multi-cœurs, puces GPGPU, *etc.*).

De plus, le traitement indépendant des disques permet la conception *d'algorithmes dynamiques*. On parle d'algorithmes dynamiques lorsque le jeu de données en entrée varie au fil du temps. Des changements fréquents exigent un recalculé systématique extrêmement rapide. Pour y parvenir, le

recalculé doit d'une manière ou d'une autre exploiter ce que l'on appelle la cohérence temporelle, c'est-à-dire les similitudes et/ou invariances par rapport à la dernière solution calculée.

Appliqué à notre contexte, il pourrait s'agir de garder en mémoire l'ensemble des configurations autorisées pour une union de disques *déformable* (comme dans une application interactive où l'utilisateur dessine une région pour optimiser manuellement certains critères). Par opposition, il pourrait aussi être question d'une forme à placer rigide mais avec des points qui varient (comme dans le cas d'un robot mobile n'ayant pas accès initialement à toute l'information sur les obstacles).

Grâce à la notion de zone interdite élémentaire introduite dans ce mémoire (notion qui ne dépend que d'un point et d'un disque), la mise à jour d'une solution de placement en fonction des disques ou points modifiés, introduits ou supprimés, peut être réalisée facilement sans refaire tous les calculs réalisés auparavant.

En conservant cette représentation par union de disques, nous aimerions pousser un peu plus loin des recherches qui font la synthèse de nos travaux, en considérant des problèmes de placement très particuliers, dont la résolution équivaut au calcul de régions de VPUD. Une telle approche, bien qu'elle exigerait un raisonnement géométrique poussé et de nombreuses preuves, pourrait déboucher sur un *calcul exact* du diagramme de VPUD. La précision d'un tel calcul serait alors bien supérieure à celle des algorithmes d'approximation présentés dans la section 4.4. Nous détaillons un peu plus cette piste de recherche en annexe A.7.

À côté des développements possibles de nos propres travaux, d'autres pistes de recherche concernent l'application de nos algorithmes actuels.

Le diagramme de VPUD peut par exemple aider à la résolution de problèmes de type : *largest empty shape*. Dans la littérature, le *largest empty circle problem* a été résolu par Toussaint en utilisant les propriétés du diagramme de Voronoï. De nombreux problèmes analogues (largest empty square, rectangle, annulus, *etc.*) ont été étudiés par la suite. Les solutions proposées n'ont toutefois pas conservé l'efficacité et l'élégance de la solution initiale de Toussaint, le diagramme de Voronoï n'étant plus adapté. De manière générale, déterminer le meilleur placement d'une forme géométrique complexe parmi des obstacles, afin de maximiser son aire, est un problème ouvert. Une solution unifiée est maintenant envisageable grâce à notre définition des régions de VPUD.

Par ailleurs, la résolution efficace de problèmes de placement et une représentation machine *ad hoc* de l'ensemble des configurations autorisées pourrait fournir un outil pour accélérer la résolution de problèmes du type *détection de collision*.

Supposons que l'on souhaite déplacer en temps réel un objet virtuel dans un environnement avec obstacles. À condition que les obstacles soient des éléments statiques, le pré-calcul des solutions au problème de placement réduirait nettement le temps de calcul *online* des collisions. Pour ce type de calcul, nos méthodes pour déterminer les configurations autorisées et l'encodage de celles-ci à l'aide de structures de données efficaces permettrait de réaliser des économies de temps de calcul indépendamment du matériel.

Annexes

A

Sommaire

A.1 Preuve de la finitude des points de jonction 151
A.2 Existence de cas particuliers simplifiant la frontière de sur-régions élémentaires 151
A.3 Preuve de la nature quartique des courbes type $\mathcal{C}'$ . . . 155
A.4 Preuve de la non-connexité d'une région de VPUD . . . 156
A.5 Implémentation de la résolution du placement par homothétie 159
A.6 Justification de l'interprétation des zones interdites, dans le contexte du PP par homothétie-rotation combinées 162
A.7 Perspectives quant au calcul exact d'une région de VPUD 165

A.1 Preuve de la finitude des points de jonction

Au chapitre 4 nous avons énoncé la propriété 4.2.2 : *finitude des points de jonction*. Nous la rappelons ci-dessous avant d'en donner la démonstration.

Finitude des points de jonction. Soient P un nuage de points et B un EDS. Pour toute région de VPUD notée $V_B(p_i)$, les points de jonction (terme introduit par la définition 4.2.1) sur sa frontière existent en nombre fini.

Démonstration. Montrons pour commencer qu'étant donnée un ensemble de sur-régions élémentaires (régions étant toutes décrites par un système d'inéquations caractéristique d'une région de VPUD), il n'existe pas deux courbes-frontières confondues.

On rappelle que les courbes-frontières sont des courbes planes de type $(\mathcal{C})$, ce type de courbe a été introduit via l'équation 4.1. Cette équation possède un certain nombre de coefficients basés sur p_i, p_j, b_k *i.e.* des éléments de P, P, B respectivement. Or, d'après la définition des EDS il n'existe pas deux éléments $b_k, b'_k \in B$ ayant des caractéristiques de centre et de rayon identiques. D'autre part, il n'existe pas deux points confondus dans P car P est un ensemble fini de points. Les coefficients de l'équation cartésienne pour deux courbes-frontières étant différents, de telles courbes sont nécessairement non confondues.

Continuons en appliquant le classique théorème de Bezout concernant les courbes planes à nos travaux, pour borner le nombre de points d'intersection entre deux courbes-frontières. Dans notre contexte, le théorème de Bezout s'énonce comme suit : deux courbes algébriques planes de degrés respectifs m, n définies sur $\mathbb{R}^2$ ont exactement $m \times n$ points d'intersections, comptés avec leur multiplicité.

Dans ce mémoire, nous avons montré qu'une courbe-frontière est une partie de courbe algébrique (propriété 4.2.4). De plus, nous avons affirmé que la courbe algébrique en question est de degré 4 (propriété 4.2.5), c'est une *quartique*. Par le théorème de Bezout, on sait que deux quartiques auront 16 points d'intersection comptés avec leur multiplicité. Toute courbe-frontière étant une partie de quartique, le nombre de points d'intersection de deux courbes-frontières est inférieur ou égal à 16.

En résumé : deux courbes-frontières d'une région de VPUD étant deux à deux non confondues, faisant toutes deux respectivement partie d'une certaine courbe algébrique de degré 4, l'utilisation du théorème de Bezout permet de conclure : les points de jonction existent en nombre fini. □

A.2 Existence de cas particuliers simplifiant la frontière de sur-régions élémentaires

Au chapitre 4, nous avons introduit les sur-régions élémentaires (définition 4.1.3) et avons étudié leurs frontières. Ces frontières sont des courbes planes dont l'équation cartésienne a été déterminée, nous l'avons développé pour aboutir à la propriété 4.2.3.

Rappelons ici cette propriété : La frontière de toute sur-région élémentaire

$S_{b_k}(p_i, p_j)$, est décrite par une équation cartésienne implicite de la forme :

$$\begin{aligned}\mathcal{C} : \left(X^2 + Y^2\right) & \frac{\gamma^2 + \delta^2 + \epsilon}{2} \\ & + \sqrt{X^2 + Y^2}\left(\gamma X + \delta Y + \alpha\gamma + \beta\delta\right) \\ & + \left(\alpha X + \beta Y + \frac{\alpha^2 + \beta^2}{2}\right) = 0 \end{aligned} \tag{A.1}$$

Où $\alpha, \beta, \gamma, \delta, \epsilon$ sont des constantes calculées à partir des données p_i, p_j, c_k, r_k. Plus précisément :

$$\begin{array}{ll} (\alpha, \beta) & = p_i - p_j \\ (\gamma, \delta) & = c_k \\ \epsilon & = 1 - r_k^2 \end{array}$$

Dans cette équation, le membre de gauche est une *somme de trois termes*. Ces termes sont déterminés par les données p_i, p_j, c_k, r_k. Sans prétendre à l'exhaustivité, nous avons mis en évidence des jeux de données particuliers qui rendent la courbe-frontière plus simple. En effet, certaines valeurs pour p_i, p_j, c_k, r_k font que un ou plusieurs termes de l'équation $\mathcal{C}$ s'annulent. La courbe ainsi modélisée est alors beaucoup plus simple.

L'objet de cette section est de présenter deux situations où la courbe se simplifie : le cas où la courbe devient une droite, et le cas où la courbe devient l'union de deux demi-droites reliées par un point.

Nous exposons ces deux cas cas sous forme de propriété d'existence, en exhibant juste un exemple simplifiant effectivement la courbe. Les hypothèses nécessaires et suffisantes à la simplification ne sont pas connues à l'heure actuelle. Chaque propriété est illustrée par un graphe réalisé à l'aide du logiciel de calcul *Mathematica*.

Propriété A.2.1 (existence du cas où une courbe type $\mathcal{C}$ est une droite)**.** *Pour $b_k = b_1 = b(\mathcal{O}, 1)$ et deux germes distincts p_i, p_j arbitrairement choisis, l'équation A.1 se simplifie pour décrire une droite :*

$$\langle x - p_i; p_i - p_j \rangle + \frac{\left\|p_i - p_j\right\|^2}{2} = 0$$

Démonstration. Par définition des ensembles de disques standard, si B est un EDS alors le disque unité placé à l'origine $b(\mathcal{O}, 1)$ fait partie de B. Supposons que le disque b_k considéré durant le calcul des sur-régions élémentaires est justement égal à $b(\mathcal{O}, 1)$. Par conséquent, les deux premiers termes de l'équation A.1 s'annulent, celle-ci devient :

$$\begin{aligned} \|x - p_i\|^2 . \frac{0}{2} + \|t\| .0 + \langle x - p_i; p_i - p_j \rangle + \frac{\left\|p_i - p_j\right\|^2}{2} &= 0 \\ \Leftrightarrow \langle x - p_i; p_i - p_j \rangle + \frac{\left\|p_i - p_j\right\|^2}{2} &= 0 \end{aligned}$$

La courbe ainsi obtenue consiste en une droite. On retrouve en fait le bissecteur entre les points p_i et p_j ; la frontière d'un demi-plan composant la région de Voronoï $V(p_i)$ classique. □

Ainsi, une courbe de type $\mathcal{C}$ peut prendre la forme d'une simple droite, auquel cas on retrouve un bissecteur associé aux régions de Voronoï classique (la frontière d'un des demi-plans composant une région de Voronoï classique). Ceci est illustré par la figure A.1a. Voyons un autre cas de figure.

Propriété A.2.2 (existence du cas où une courbe type $\mathcal{C}$ est l'union de deux demi-droites)**.**

Démonstration. Nous partons de l'équation caractérisant la frontière d'une sur-région élémentaire :

$$\|x-p_i\|^2\left(\frac{1+\|c_k\|^2-r_k^2}{2}\right)+\|t\|\langle c_k;x-p_j\rangle + \langle x-p_i;p_i-p_j\rangle+\frac{\|p_i-p_j\|^2}{2}=0$$

Sous condition d'avoir $p_j = p_i$, les troisième et quatrième termes de la somme (membre de gauche) s'annulent. L'équation de la courbe devient :

$$\|x-p_i\|^2\left(\frac{1+\|c_k\|^2-r_k^2}{2}\right)+\|x-p_i\|\langle c_k;x-p_i\rangle=0 \qquad \text{(A.2)}$$

On énumère deux cas :

- si $x = p_i$ auquel cas $x - p_i = 0$ et l'équation est satisfaite.
- sinon, on a $\|x - p_i\| > 0$ obligatoirement, et l'équation A.2 est équivalente à :

$$\|x-p_i\|\left[\|x-p_i\|\left(\frac{1+\|c_k\|^2-r_k^2}{2}\right)+\langle x-p_i;c_k\rangle\right]=0$$

$$\Leftrightarrow \|x-p_i\|\left(\frac{1+\|c_k\|^2-r_k^2}{2}\right)+\langle x-p_i;c_k\rangle=0 \qquad \text{(A.3)}$$

Nous exhibons à présent un exemple concret simplifiant davantage la courbe. Considérons le jeu de données suivant :

$$\begin{cases} p_i &= (0,0)\\ c_k &= (1,0)\\ r_k &= \frac{1}{2}\end{cases}$$

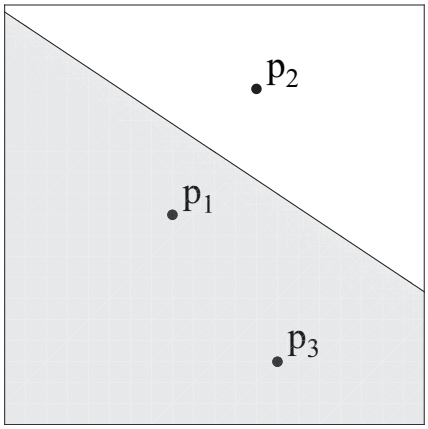
Dans ce cas, en notant (X, Y) les coordonnées du point $(x - p_i)$, l'équation A.3 devient :

$$\|x-p_i\|\left(\frac{1+1-\frac{1}{4}}{2}\right)+\langle x-p_i;(1,0)\rangle=0 \Leftrightarrow \frac{7}{8}\|x-p_i\|+X=0$$

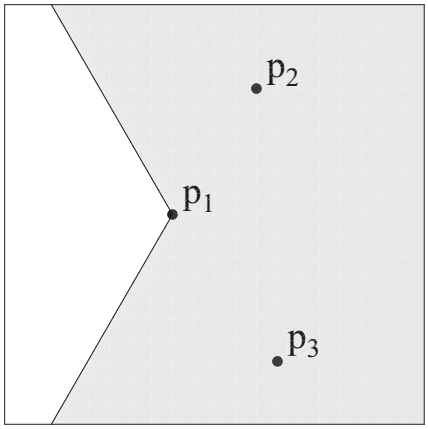
$$\Leftrightarrow \frac{7}{8}\sqrt{X^2+Y^2}+X=0$$

Trois cas se présentent concernant la coordonnée X. Soit $X = 0$ auquel cas le point $(0, 0)$ est la seule solution de l'équation. Soit $X > 0$ auquel cas l'équation n'a aucune solution. Soit $X < 0$ auquel cas l'équation devient :

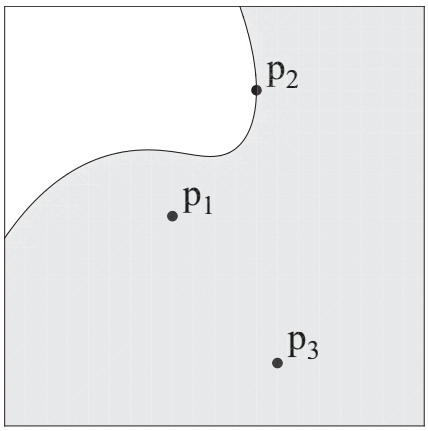
$$\frac{7}{8}\sqrt{X^2+Y^2}-\mid X\mid=0$$



(a) *Cas particulier $k = 1$. La droite obtenue est un des bissecteurs délimitant la région de Voronoï classique*



(b) *Cas particulier $p_j = p_i$. La courbe obtenue consiste en deux demi-droites d'extrémité p_i*



(c) *Autres cas. La courbe-frontière étudiée est non linéaire*

FIGURE A.1 – *Cas particuliers concernant la courbe délimitant une sur-région élémentaire*

Par suite, trois cas se présentent concernant la coordonnée Y. Soit $Y = 0$ auquel cas l'équation résultante $\frac{7}{8} \mid X \mid - \mid X \mid = 0$ n'a pas de solutions. Soit $Y > 0$ auquel cas l'équation devient :

$$\begin{aligned}
\frac{7}{8}\sqrt{X^2+Y^2} = \mid X \mid &\Leftrightarrow \left(\frac{7}{8}\right)^2 (X^2+Y^2) = \mid X \mid^2 \\
&\Leftrightarrow \left(\frac{7}{8}\right)^2 Y^2 = \mid X \mid^2 \left(1 - \left(\frac{7}{8}\right)^2\right) \\
&\Leftrightarrow \frac{7}{8} \mid Y \mid = \sqrt{1 - \left(\frac{7}{8}\right)^2} \mid X \mid \\
&\Leftrightarrow \frac{7}{8} Y = -X\sqrt{1 - \left(\frac{7}{8}\right)^2} \\
&\text{d'après nos hypothèses sur les signes respectifs de } X, Y \\
&\Leftrightarrow X\sqrt{1 - \left(\frac{7}{8}\right)^2} + \frac{7}{8} Y = 0
\end{aligned}$$

Soit $Y < 0$ auquel cas l'équation devient :

$$X\sqrt{1 - \left(\frac{7}{8}\right)^2} - \frac{7}{8} Y = 0$$

Pour résumer, on a observé que le point $p_i = (0,0)$ est solution de l'équation, que pour $X < 0$ et $Y > 0$ l'équation devient l'équation d'une droite, que pour $X < 0$ et $Y < 0$ l'équation devient l'équation d'une droite différente de la première. Finalement, pour le jeu de données choisi, la courbe-frontière est l'union de deux demi-droites ayant leur extrémité p_i comme seul point commun. □

Cette deuxième propriété est illustrée sur la figure A.1b.

A.3 Preuve de la nature quartique des courbes type $\mathcal{C}'$

Au chapitre 4 nous avons mentionné deux types de courbes planes : les courbes type $\mathcal{C}$ et les courbes type $\mathcal{C}'$. De telles courbes sont distinguables par leur équation cartésienne (les équations cartésiennes en question ont été données dans les propriétés 4.2.3, 4.2.4 respectivement).

La propriété 4.2.5 a assimilé les courbes type $\mathcal{C}'$ à des courbes algébriques planes de degré 4 *i.e.* des quartiques. Nous rappelons ici l'énoncé de cette propriété avant d'en donner une démonstration.

Nature quartique des courbes type $\mathcal{C}'$. Toute courbe dont l'équation cartésienne est de la forme :

$$\begin{aligned}
\mathcal{C}' : &- (ac + bd + cX + dY)^2 \left(X^2 + Y^2\right) \\
&+ \left(\frac{1}{2}\left(a^2 + b^2\right) + aX + bY + \frac{1}{2}\left(c^2 + d^2 + e\right)\left(X^2 + Y^2\right)\right)^2 = 0
\end{aligned}$$

est en fait une courbe algébrique de degré 4 (une quartique). Autrement dit, toute courbe type C' peut aussi s'écrire sous la forme :

$$C' : Ax^4 + By^4 + Cx^3y + Dx^2y^2 + Exy^3 + Fx^3 + Gy^3 + Hx^2y + Ixy^2 + Jx^2 + Ky^2 + Lxy + Mx + Ny + O = 0$$

Démonstration. La démonstration consiste à développer entièrement la première forme d'équation cartésienne. Pour simplifier légèrement, on multiplie les deux membres de l'équation par 4, cela permet de retirer de nombreux coefficients $\frac{1}{4}$ qu'il faudrait conserver tout au long du développement dans le cas contraire. Le calcul étant extrêmement long, nous préférons ne le reproduire en intégralité ici. Nous avons vérifié une à une les étapes de calcul à l'aide d'un logiciel de calcul formel (*Mathematica*). Après développement des termes de l'équation, on obtient :

$$\begin{aligned}
C' :& a^4 + 2a^2b + b^4 + 4a^3x + 4ab^2x + 4a^2x^2 \\
&- 2a^2c^2x + 2b^2c^2x^2 - 8abcdx^2 + 2a^2d^2x^2 - 2b^2d^2x^2 \\
&+ 2a^2ex^2 + 2b^2ex^2 - 4ac^2x^3 - 8bcdx^3 + 4ad^2x^3 + 4aex^3 \\
&- 4c^2x^4 + c^4x^4 + 2c^2d^2x^4 + d^4x^4 \\
&+ 2c^2ex^4 + 2d^2ex^4 + e^2x^4 + 4a^2by + 4b^3y + 8abxy \\
&+ 4bc^2x^2y - 8acdx^2y - 4bd^2x^2y \\
&+ 4bex^2y - 8cdx^3y + 4b^2y^2 - 2a^2c^2y^2 + 2b^2c^2y^2 - 8abcdy^2 \\
&+ 2a^2d^2y^2 - 2b^2d^2y^2 \\
&+ 2a^2ey^2 + 2b^2ey^2 - 4ac^2xy^2 - 8bcdx^2y^2 + 4ad^2xy^2 + 4aexy^2 \\
&- 4c^2x^2y^2 + 2c^4x^2y^2 \\
&- 4d^2x^2y^2 + 4c^2d^2x^2y^2 + 2d^4x^2y^2 + 4c^2ex^2y^2 + 4d^2ex^2y^2 + 2e^2x^2y^2 \\
&+ 4bc^2y^3 - 8acdy^3 \\
&- 4bd^2y^3 + 4bey^3 - 8cdxy^3 + c^4y^4 - 4d^2y^4 + 2c^2d^2y^4 \\
&+ d^4y^4 + 2c^2ey^4 + 2d^2ey^4 + e^2y^4 = 0
\end{aligned}$$

Finalement, en factorisant selon les puissances identiques de x, y et en substituant aux expressions en facteur les coefficients $A, B, C, \ldots, N$ comme définis dans la propriété, on retombe bien sur le résultat attendu. Toute courbe de type C' est donc une quartique. □

A.4 Preuve de la non-connexité d'une région de VPUD

Au chapitre 4 nous avons étudié les propriétés des régions de VPUD. Par la propriété 4.3.4, nous avons énoncé le fait qu'une région de VPUD ne consiste pas nécessairement en une région du plan connexe. Rappelons ci-dessous l'énoncé de cette propriété avant d'en donner une démonstration.

Non-connexité d'une région de VPUD. Il existe au moins un triplet : nuage de points P, germe p_i, ensemble de disques standard B, tel que la région $V_B(p_i)$ soit non connexe. Formellement :

$$\exists A_1, A_2 \subset \mathbb{R}^2; A_1 \cap A_2 = \emptyset \text{ et } [A \cap V_B(p_i)] \cup [B \cap V_B(p_i)] = V_B(p_i)$$

Autrement dit, pour certains jeux de données une région de VPUD est décomposable en plusieurs composantes disjointes.

Démonstration. La démarche employée pour prouver ce résultat est la suivante. Nous commençons par introduire un jeu de données P, B particulier et considérons un certain germe p_i. Nous montrons alors qu'il existe deux points t_1, t_2 du plan appartenant tous deux à la région $V_B(p_i)$, mais situés de part et d'autre d'une droite (D). Nous montrons finalement qu'aucun point de (D) n'appartient à la région $V_B(p_i)$.

Le jeu de données que l'on considère pour établir la propriété est le suivant :

$$\begin{array}{lllllllll} P & = & \{p_1, p_2\} & B & = & \{b(c_1, r_1), b(c_2, r_2)\} & & & \\ p_1 & = & (0,0) & c_1 & = & (0,0) & \text{et } r_1 & = & 1 \\ p_2 & = & (\frac{1}{2}, 2) & c_2 & = & (1,0) & \text{et } r_2 & = & 1 \end{array}$$

Pour ce jeu de données, l'union de disques $\cup B$ est l'union de deux disques de rayon unité placés sur un même axe horizontal. Dans la suite, on s'intéresse uniquement à la région liée au germe p_1, région notée $V_B(p_1)$.

À présent, considérons deux points du plan euclidien : $t_1 = (-4, 0)$ et $t_2 = (4, 0)$ et vérifions qu'ils appartiennent à la région $V_B(p_1)$. Pour ce faire, on utilise la caractérisation algébrique des région de VPUD, caractérisation donnée par le théorème 4.1.6. Appliqué à notre cas, ce théorème dit que la région $V_B(p_1)$ est égale aux solutions du système algébrique suivant :

$$\begin{cases} \|c_1(a) - p_1\| \geq r_1(a) \\ \|c_2(a) - p_1\| \geq r_2(a) \\ \|c_1(a) - p_2\| \geq r_1(a) \\ \|c_2(a) - p_2\| \geq r_2(a) \end{cases}$$

Où $a = (x, \|x - p_1\|, 0)$ d'après la définition 3.3.3 des régions de VPUD. Notre choix des données P, B fait que nous n'avons que quatre inéquations à considérer, chaque inéquation du système correspondant à une combinaison : élément de P, élément de B. Développons ces inéquations une à une, en remplaçant chaque terme par sa valeur connue (*application numérique*) :

1. L'inéquation correspondant au couple p_1, b_1 devient :

$$\begin{aligned} \|t + \|t - p_1\| \, c_1 - p_1\| &\geq \|t - p_1\| \, r_1 \\ \Leftrightarrow \|t + 0 - p_1\| &\geq \|t - p_1\| \, 1 \\ \Leftrightarrow \|t - p_1\| &\geq \|t - p_1\| \end{aligned}$$

   cette équation étant toujours vraie, on l'ignore dans la suite.

2. L'inéquation correspondant au couple p_1, b_2 devient :

$$\begin{aligned} \|t + \|t - p_1\| \, c_2 - p_1\| &\geq \|t - p_1\| \, r_2 \\ \Leftrightarrow \|t + \|t\| \, c_2\| &\geq \|t\| \, r_2 \\ \Leftrightarrow x^2 + y^2 + (x^2 + y^2) + 2y\sqrt{x^2 + y^2} &\geq x^2 + y^2 \\ \Leftrightarrow x^2 + y^2 + 2y\sqrt{x^2 + y^2} &\geq 0 \qquad \text{(ineq2)} \end{aligned}$$

3. L'inéquation correspondant au couple p_2, b_1 devient :

$$\begin{aligned} \|t + \|t - p_1\| c_1 - p_2\| &\geq \|t - p_1\| r_1 \\ \Leftrightarrow \|t - p_2\| &\geq \|t\| \\ \Leftrightarrow \left(x - \frac{1}{2}\right)^2 + (y-2)^2 &\geq x^2 + y^2 \\ \Leftrightarrow \frac{1}{4} - x + 4 - 4y &\geq 0 \end{aligned} \tag{ineq3}$$

4. L'inéquation correspondant au couple p_2, b_2 devient :

$$\begin{aligned} \|t + \|t - p_1\| c_2 - p_2\| &\geq \|t - p_1\| r_2 \\ \Leftrightarrow \|t - p_2\|^2 + \|t\|^2 \|c_2\|^2 + 2\langle t - p_2; \|t\| c_2\rangle &\geq \|t\|^2 \\ \Leftrightarrow (x - \frac{1}{2})^2 + (y-2)\left[(y-2)\sqrt{x^2+y^2}\right] &\geq 0 \end{aligned} \tag{ineq4}$$

Pour vérifier l'appartenance de $t_1 = (-4, 0)$ à la région $V_B(p_1)$ nous substituons ses coordonnées à x. Par le calcul l'inéquation ineq2 devient $16 \geq 0$, l'inéquation ineq3 devient $\frac{1}{4} + 8 \geq 0$, l'inéquation ineq4 devient $\frac{81}{4} - \frac{16}{4} \geq 0$ Toutes ces inéquations étant vraies, t_1 appartient bien à la région $V_B(p_1)$. On procède de la même manière pour vérifier l'appartenance de $t_2 = (4, 0)$ à la région $V_B(p_1)$. Par le calcul l'inéquation ineq2 devient aussi $16 \geq 0$, l'inéquation ineq3 devient $\frac{1}{4} \geq 0$, l'inéquation ineq4 devient $\frac{49}{4} - \frac{48}{4} \geq 0$ Toutes ces inéquations étant vraies, t_2 appartient bien à la région $V_B(p_1)$. On note (D) la droite verticale d'équation $x = 2$. Pour finir cette étape de la démonstration, remarquons que les points t_1, t_2 se trouvent de part à d'autre de (D).

Pour finir, montrons qu'il n'existe aucun point de la droite (D) appartenant à la région $V_B(p_1)$. En remplaçant la coordonnée x par 2 dans l'inéquation ineq3, celle-ci devient :

$$\frac{9}{16} - y \geq 0$$

Une telle fonction en y est monotone et strictement décroissante. Elle est donc strictement négative sur l'intervalle $y \in]\frac{9}{16}, +\infty[$. Une inéquation fausse suffit à invalider l'hypothèse d'appartenance du point à la région, car pour qu'un point appartienne à la région il est nécessaire que *toutes* les inéquations soient vérifiées. Tout point candidat de la forme $(x, y) = (2, y')$ avec $y' > \frac{9}{16}$ n'appartient pas à la région de VPUD $V_B(p_1)$. Un raisonnement analogue est requis pour prouver la non-appartenance de points candidat de la forme $(x, y) = (2, y'')$ avec $y'' \leq \frac{9}{16}$. Comme les inéquations ineq2 et ineq4 sont plus complexes, le raisonnement en question consiste à déterminer les signes et variations de fonctions qui sont non-linéaires. Bien qu'il ne semble ne pas y avoir d'obstacle théorique, la longueur des calculs complexifie considérablement la démonstration. Dans le cadre de cette thèse avons supposé la véracité de ce résultat que nous avons testé numériquement. Aucun des résultats de calcul obtenus n'a contredit cette intuition à propos de la non-appartenance des points de (D) à la région $V_B(p_1)$. On conjecture ainsi qu'aucun point de la droite (D) n'appartient à la région de VPUD associée au germe p_1.

Au vu des trois faits mentionnés :

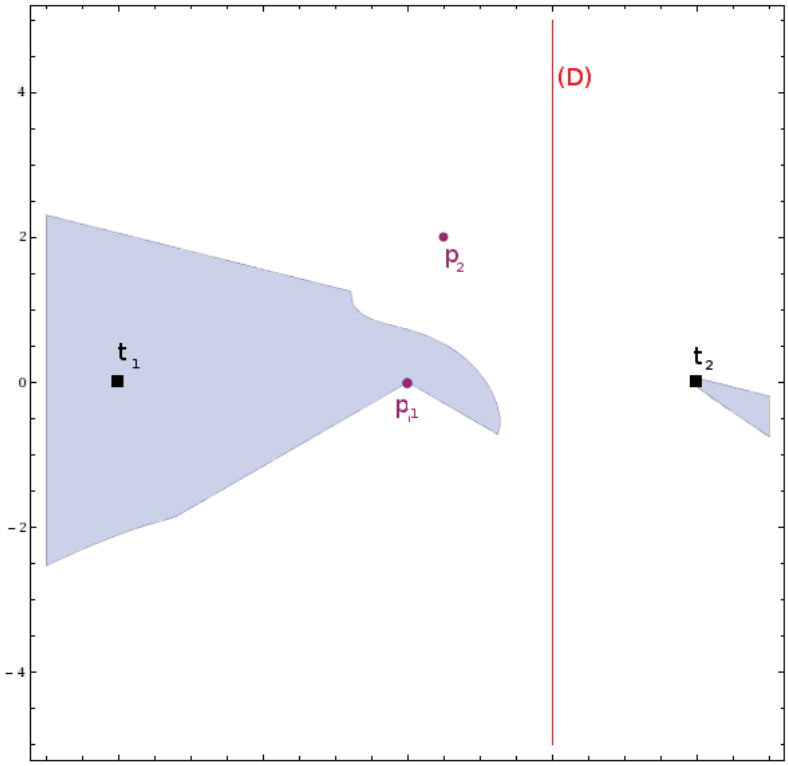


FIGURE A.2 – *Exemple de région de VPUD non-connexe (en bleu)*

– Les points t_1 et t_2 appartiennent à la région $V_B(p_1)$
– Les points t_1, t_2 se situent de part et d'autre d'une droite (D) d'équation $x = 2$
– Tout point choisi sur la droite (D) n'appartient pas à la région $V_B(p_1)$

Nous concluons que pour le jeu de données considéré, la région de VPUD associée à p_1 que l'on note $V_B(p_1)$ est nécessairement non connexe. Dit autrement, une région de VPUD peut consister en au moins deux régions du plan disjointes. □

La figure A.2 illustre ce contre-exemple de la connexité. Les points t_1, t_2 utilisés dans la preuve sont représentés en bleu. On voit qu'ils se trouvent bien de part et d'autre de la droite (D). Ce contre-exemple a été construit de sorte à être aussi simple que possible, il en existe d'autres. L'ensemble de disques considéré est représenté à part, sur la figure A.3.

A.5 Implémentation de la résolution du placement par homothétie

Au chapitre 6, nous avons proposé une méthode générique de placement d'union de disques dans un nuage de points. Pour illustrer son fonctionnement en pratique, nous avons appliqué cette méthode à différents problèmes de placement (ou PP).

Dans la sous-section 6.3.1 en particulier, la résolution du PP par homothétie uniquement a été détaillée. Rappelons que ce problème prend en entrée un nuage de points P, un EDS B, un point de référence x_k et une

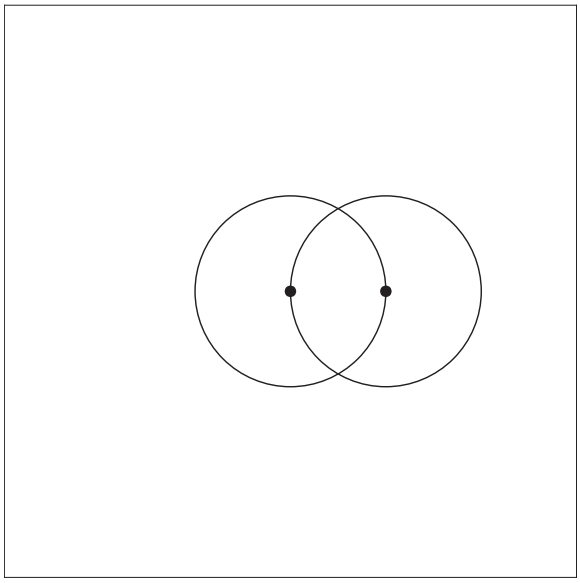


FIGURE A.3 – *Ensemble de disques utilisé pour la preuve de non-connexité*

orientation fixée θ_k. L'espace de configurations à considérer est défini par :

$$E = E_{H.} := x_k \times \mathbb{R}^+ \times \theta_k$$

Les définitions d'orbite et de zone interdite avaient été analysées dans ce contexte-là. Nous avions conclu que les orbites consistent alors en des demi-droites ayant pour extrémité x_k, et qu'une *zone interdite élémentaire* consiste soit en l'ensemble vide, soit en un segment de droite, soit en une demi-droite. Pour déterminer dans quel cas on se trouve, on utilise le fait qu'une zone interdite élémentaire pour un couple $b_i \in B, p_j \in P$ peut être décrite comme l'ensemble des points membres de l'orbite $orb_i(E_{H.})$ et qui vérifient l'inéquation :

$$\lambda^2 \left(\|c_j\|^2 - r_j^2 \right) + \lambda 2 \langle u_k - p_i; \mathbf{M}_{\theta_k} c_j \rangle + \|u_k - p_i\|^2 < 0$$

Dans cette section nous présentons un extrait de code directement issu de notre implémentation (langage de programmation utilisé : *Python 2.6*), pour donner un aperçu de la gestion des différents cas, basée sur la fonction polynomiale :

$$g(\lambda) = \lambda^2 \left(\|c_j\|^2 - r_j^2 \right) + \lambda 2 \langle u_k - p_i; \mathbf{M}_{\theta_k} c_j \rangle + \|u_k - p_i\|^2$$

Ci-dessous se trouve le listing d'instructions définissant la classe `ZoneInterditeElementaire` qui modélise une zone interdite élémentaire dans le cas du PP par homothétie.

```
# -*- coding: utf-8 -*-
#  classes pour le calcul d'une zone interdite élémentaire
#  dans le contexte du placement par homothétie

from math import cos, sin, sqrt
```

```
class IntervalOuvert:
    #classe modélisant un intervalle réel

    def __init__(self, binf, bsup ):
        #constructeur
        self.binf, self.bsup = binf, bsup
        if self.binf > self.bsup:
            self.binf, self.bsup = self.bsup, self.binf

    def __str__(self):
        # fonction d'affichage de l'intervalle
        res = ''
        res+="] "
        res+= ( str(self.binf) + "; " + str(self.bsup) )
        res+=" ["
        return res

class ZoneInterditeElementaire:

    def __init__(self, thetaK, xK, vCi, ri, ptPj):
    # thetaK et xK sont les constantes du pb.
        (xv,yv) = map(float, xK )
        (xp, yp) = map(float, ptPj )
        (xci, yci) = map(float, vCi )
        ri = float(ri )

        #calcul des coefficients polynomiaux
        cA = xci**2 + yci**2 - ri**2
        cB = (-2) * (
            (xp-xv)*( xci * cos(thetaK) - yci * sin(thetaK) )
            + (yp-yv)*( xci * sin(thetaK) + yci * cos(thetaK) )
            )
        cC = (xp-xv)**2 + (yp-yv)**2

        def g(lambda):
        #fonction polynomiale utilisée pour
#calculer l'étendue de la zone interdite
            return (cA*(lambda**2) + cB*lambda + cC )

        #variable pour stocker les racines de l'équation g(lambda)=0
        self.contenu = []

        if (cA!=0):
            #si changement de signe, comme g(0) positif ou nul,
#alors zone interdite = ens.vide
            DISCRIM = cB**2 -4*cA*cC
            if DISCRIM > 0:
                #il existe deux solutions
                s1 = ( -cB - sqrt(DISCRIM) ) / (2*cA)
                s2 = ( -cB + sqrt(DISCRIM) ) / (2*cA)
                if s2<s1:
                    s2,s1=s1,s2
                mid = (-cB)/(2*cA)
                if kappa(mid)<0:
                    self.contenu.append( IntervalOuvert( s1, s2 ) )
                else:
                    self.contenu.append( IntervalOuvert( float('-inf'), s1 ) )
```

```
                    self.contenu.append( IntervalOuvert( s2, float('+inf') ) )
            else:
                #pas de solution pour g(lambda)<0
                pass
        elif (cB!=0):
            if(cB>0):
                self.contenu.append( IntervalOuvert( float('-inf'), -cC/cB) )
            else:
                self.contenu.append( IntervalOuvert( -cC/cB, float('inf') ) )
        else:
            #pas de solution pour g(lambda)<0
            pass
        #fin de constructeur
#fin de définition de classe
```

Remarque. Le stockage du nombre de solutions n'est pas nécessaire en langage *Python*. Celui-ci a une syntaxe et des « *built-in functions* » très pratiques qui permettent d'écrire, par exemple : `len(self.contenu)` pour accéder à la longueur de la liste nommée `contenu` qui est membre de l'objet instancié courant. De plus, les blocs d'instructions sont délimités directement suivant l'indentation et non avec des accolades comme en C.

A.6 Justification de l'interprétation des zones interdites, dans le contexte du PP par homothétie-rotation combinées

Au chapitre 6, après avoir présenté la définition 6.2.4 de zone interdite dans le cas général, nous avons étudié le cas du placement par homothétie-rotation combinées. La définition de zone interdite se simplifie alors, on peut en donner une interprétation géométrique, ce qui a été fait avec la propriété 6.3.1. Rappelons cette propriété avant de la justifier en détail.

Interprétation des zones interdites, pp homothétie-rotation combinées Toute zone interdite élémentaire $\delta_{i,p_j}(E_{R.H.})$, dans le cas du placement par homothétie-rotation combinées n'est composée que de disques et de demi-plans. En effet :

- si $\|c_i\| > r_i$ alors A est égal à un disque ouvert $b(C,R)$ où : $C = \left(\frac{\|c_i\|^2 p_x - r_i^2 v_x}{\|c_i\|^2 - r_i^2}, \frac{\|c_i\|^2 p_y - r_i^2 v_y}{\|c_i\|^2 - r_i^2}\right)$ et $R = \|p - v\|^2 \left(\frac{\|c_i\| r_i}{\|c_i\|^2 - r_i^2}\right)$
- si $\|c_i\| = r_i$ alors A est égal à un demi-plan de vecteur directeur $\overrightarrow{x_k p}$ et passant par le point O situé à égale distance de p et x_k tel que : $O = (p + x_k)/2$
- si $\|c_i\| < r_i$ alors A est égal au complément du disque fermé $\overline{b}(C,R)$ où C et R sont définis comme précédemment.

Démonstration. Dans le cas du placement par homothétie-rotation combinées, le fait que $z \in \delta_{i,p_j}(E_{R.H.})$ est équivalent à :

$$z \in orb_i(E_{R.H.}) \text{ et } \left\|p_j - z\right\| < \|u_k - z\| \frac{r_i}{\|c_i\|} \tag{A.4}$$

Posons : $\alpha_i = \frac{r_i}{\|c_i\|}$. Nous remplaçons par ce terme, et puisque les deux membres de l'inégalité A.4 sont positifs, celle-ci équivaut à :

$$\begin{aligned} &\left\|z - p_j\right\|^2 < {\alpha_i}^2 \left\|u_k - z\right\|^2 \\ \Leftrightarrow\ &\|z\|^2 + \left\|p_j\right\|^2 - 2\langle z; p_j\rangle - {\alpha_i}^2 \left(\|z\|^2 + \|u_k\|^2 - 2\langle z; u_k\rangle\right) < 0 \\ \Leftrightarrow\ &\left(1 - \alpha_i^2\right)\|z\|^2 + 2\alpha_i^2\langle z; u_k\rangle - 2\langle z; p_j\rangle + \left\|p_j\right\|^2 - \alpha_i^2\|u_k\|^2 < 0 \end{aligned}$$

Nous décomposons davantage, en adoptant la notation suivante pour les coordonnées : $z(x, y)$ et $u_k(u_{k,x}, u_{k,y})$ et $p_j(p_{j,x}, p_{j,y})$. Ainsi, on peut réécrire la dernière inégalité sous la forme :

$$\begin{aligned} &\left(1 - \alpha_i^2\right)\left(x^2 + y^2\right) + 2\alpha_i^2\left(xu_{k,x} + yu_{k,y}\right) \\ &\qquad - 2\left(xp_{j,x} + yp_{j,y}\right) + \left\|p_j\right\|^2 - \alpha_i^2\|u_k\|^2 < 0 \end{aligned}$$

Celle-ci est de la forme :

$$A\left(x^2 + y^2\right) + Bx + Cy + D < 0$$

$$\text{avec } \begin{cases} A &= 1 - \alpha_i^2 = 1 - \frac{r_i^2}{\|c_i\|^2} \\ B &= 2\alpha_i^2 u_{k,x} - 2p_{j,x} \\ C &= 2\alpha_i^2 u_{k,y} - 2p_{j,y} \\ D &= \|p\|^2 - \alpha_i^2\|u_k\|^2 \end{cases}$$

Maintenant que les coefficients A, B, C, D ont été déterminés, nous pouvons nous passer de la notation α_i. Remplaçons α_i par la valeur correspondante. Pour chaque coefficient, on peut d'ailleurs mettre $\frac{1}{\|c_i\|}$ en facteur. Il vient que :

$$\begin{cases} A &= \frac{\|c_i\|^2 - r_i^2}{\|c_i\|^2} \\ B &= \frac{2\left(r_i^2 u_{k,x} - \|c_i\|^2 p_{j,x}\right)}{\|c_i\|^2} \\ C &= \frac{2\left(r_i^2 u_{k,y} - \|c_i\|^2 p_{j,y}\right)}{\|c_i\|^2} \\ D &= \frac{\|c_i\|^2\left\|p_j\right\|^2 - r_i^2\|u_k\|^2}{\|c_i\|^2} \end{cases}$$

Ce facteur commun $\frac{1}{\|c_i\|}$ étant strictement positif, il est possible de le supprimer de l'inégalité sans en changer le sens. L'inégalité simplifiée ainsi obtenue est :

$$A'\left(x^2 + y^2\right) + B'x + C'y + D' < 0$$

$$\text{avec } \begin{cases} A' &= \|c_i\|^2 - r_i 2 \\ B' &= 2\left(r_i^2 v_x - \|c_i\|^2 p_x\right) \\ C' &= 2\left(r_i^2 v_y - \|c_i\|^2 p_y\right) \\ D' &= \|c_i\|^2\|p\|^2 - r_i^2\|v\|^2 \end{cases}$$

À présent, pour fournir une interprétation géométrique de la zone interdite, nous distinguons trois cas suivant que A' soit nul, positif ou négatif.

– **c**as où A' est nul. Dans ce cas de figure, on sait que $\|c_i\| = r_i$ car $\|c_i\|, r_i$ sont positifs strictement. Donc la zone interdite est un demi-plan défini par l'inégalité $B'x + C'y + D < 0$. Explicitement :

$$2\left(r_i{}^2 u_{k,x} - \|c_i\|^2 p_{j,x}\right) x + 2\left(r_i{}^2 u_{k,y} - \|c_i\|^2 p_{j,y}\right) y + \|c_i\|^2 \|p_j\|^2 - r_i^2 \|u_k\|^2 < 0$$

Cette inéquation se simplifie en utilisant notre hypothèse $\|c_i\| = r_i$, elle devient :

$$2\left(r_i^2 u_{k,x} - r_i{}^2 p_{j,x}\right) x + 2\left(r_i^2 u_{k,y} - r_i{}^2 p_{j,y}\right) y + r_i{}^2 \|p_j\|^2 - r_i^2 \|u_k\|^2 < 0$$

On factorise le terme r_i pour aboutir à :

$$2r_i^2 \left[(v_x - p_x)x + (x_y - p_y)y + \frac{\|p\|^2 - \|v\|^2}{2}\right] < 0$$

Puisque $2r_i^2$ est strictement positif, ce terme peut être supprimé de l'inéquation sans en changer le sens.
La frontière de ce demi-plan est une droite. Pour terminer la description de ce demi-plan nous allons identifier à présent le vecteur directeur et le point de passe de celle-ci. Rappelons qu'en géométrique analytique, il est connu que toute droite passant par un point $M(x_M, y_M)$ de vecteur directeur $\vec{v} = (\alpha, \beta)$ admet pour équation générale :

$$\beta x - \alpha y + (\alpha y_M - \beta x_M) = 0$$

Ainsi, on procède par identification pour déterminer le vecteur directeur $\vec{v} = (\alpha, \beta)$ et le point de passage $M(x_M, y_M)$:

$$\begin{cases} \beta & = & (v_x - p_x) \\ -\alpha & = & (v_y - p_y) \\ (\alpha y_M - \beta x_M) & = & \frac{\|p\|^2 - \|v\|^2}{2} \end{cases}$$

$$\Rightarrow \begin{cases} \alpha & = & p_y - v_y \\ \beta & = & v_x - p_x \\ x_M & = & \frac{v_x + p_x}{2} \\ y_M & = & \frac{v_y + p_y}{2} \end{cases}$$

– **cas où A′ est positif.** Tout d'abord, comme $A' \neq 0$ l'inéquation A.4 se simplifie :

$$A'\left(x^2 + y^2\right) + B'x + C'y + D' < 0$$
$$\Leftrightarrow A'\left[x^2 + y^2 + \frac{B'}{A'}x + \frac{C'}{A'}y + \frac{D'}{A'}\right] < 0$$

Le fait que $A' > 0$ est équivalent à $\|c_i\| > r_i$, on peut se ramener à :

$$x^2 + y^2 + \frac{B'}{A'}x + \frac{C'}{A'}y + \frac{D'}{A'} < 0$$
$$\Leftrightarrow \left(x + \frac{B'}{2A'}\right)^2 - \left(\frac{B'}{2A'}\right)^2 + \left(y + \frac{C'}{2A'}\right)^2 - \left(\frac{C'}{2A'}\right)^2 + \frac{D'}{A'} < 0$$
$$\Leftrightarrow \left(x + \frac{B'}{2A'}\right)^2 + \left(y + \frac{C'}{2A'}\right)^2 < \left(\frac{B'}{2A'}\right)^2 + \left(\frac{C'}{2A'}\right)^2 - \frac{D'}{A'}$$

A partir de la forme de cette inéquation on peut en déduire que la zone interdite correspond à un disque ouvert ; une région bornée. On identifie aisément le centre C du disque :

$$C = \left(-\frac{B'}{2A'}; -\frac{C'}{2A'}\right) = \left(\frac{\|c_i\|^2 p_x - r_i^2 v_x}{\|c_i\|^2 - r_i^2}; \frac{\|c_i\|^2 p_y - r_i^2 v_y}{\|c_i\|^2 - r_i^2}\right)$$

D'autre part, en prenant la racine carrée du second terme de l'inéquation et en développant les calculs (on se sert aussi du fait que $r_i, \|c_i\|$ sont des réels strictement positifs) on trouve que le rayon du disque :

$$\sqrt{\frac{B'^2}{2A'} + \frac{C'^2}{2A'} - \frac{D'}{A'}} \text{ est égal à } \frac{\|c_i\| \, r_i \, \|p - v\|}{| \, \|c_i\|^2 - r_i^2 \, |}$$

– **cas où A′ est négatif.** De la même manière que pour le cas où A' est positif, le fait que A' soit non nul permet d'écrire l'inéquation sous la forme :

$$A' \left[x^2 + y^2 + \frac{B'}{A'}x + \frac{C'}{A'}y + \frac{D'}{A'}\right] < 0$$

Les calculs peuvent être développés de façon quasi-identique que pour le cas précédent (cas $A' > 0$). Le fait que $A' < 0$ est équivalent à : $\|c_i\| < r_i$. Au final, seul le signe l'inéquation obtenue change (changement de signe dû à la multiplication des deux membres de l'inéquation par un nombre négatif). On se ramène à :

$$x^2 + y^2 + \frac{B'}{A'}x + \frac{C'}{A'}y + \frac{D'}{A'} > 0$$

La zone interdite correspond alors au complément $\complement_{\mathbb{R}^2}\overline{b(C,R)}$ du disque fermé de rayon R centré en C, avec C et R identiques aux valeurs calculées pour le case précédent. Pour $A' < 0$, la zone interdite correspond donc à une région non bornée ouverte.

□

A.7 Perspectives quant au calcul exact d'une région de VPUD

Cette section présente plus en détail la piste de recherche mentionnée en conclusion de la thèse. À long terme, notre but est la conception d'un algorithme de calcul *exact* d'une région de VPUD, par opposition aux méthodes d'approximation proposées au chapitre 4. Les travaux résumés ici ont été menés *a posteriori* de notre généralisation du diagramme de Voronoï et de notre étude du placement d'union de disques. Ils empruntent des notions à ces deux thèmes et en font en quelque sorte la synthèse.

Le développement théorique pour réaliser le calcul exact n'étant pas assez avancé au moment de la rédaction du présent document, nous avons préféré exposer en annexe, de manière plus informelle, notre piste de recherche. Celle-ci consiste en quatre observations qui s'articulent les unes aux autres :

- il existe un lien « calculatoire » entre problèmes de placement. Le calcul de solutions pour un PP particulier suffit, sous certaines conditions, à résoudre un autre PP. Dans un tel cas de figure, le premier PP est dit supérieur au second ;
- il existe un lien entre les régions de Voronoï et le placement d'un disque dans un nuage de points. Par extension, il existe un lien entre les régions de VPUD et le placement d'une union de disques dans un nuage de points ;
- il existe un PP « spécial » en ceci que sa résolution correspond précisément au calcul d'une région de VPUD ;
- en considérant un PP supérieur au PP spécial, le PP spécial peut être résolu.

Ci-dessous, nous détaillons ces observations une à une, avec un paragraphe dédié pour chacune d'entre elles. Par la suite, nous proposerons un procédé pour calculer une région de VPUD sans approximation.

Lien calculatoire inter-problèmes de placement : L'étude et la résolution de PP ont été abordés aux chapitres 5 et 6. Commençons par une observation particulière : considérant le PP par rotation uniquement et le PP par homothétie-rotation combinées, les solutions $sol_{P,B}(E_{R.})$ du PP par rotation utilisant un paramètre x_k peuvent être déduites directement à partir des solutions $sol_{P,B}(E_{R.H.})$ du PP par homothétie-rotation utilisant le même paramètre x_k. Les solutions de ces deux problèmes de placement ne sont pas indépendantes l'une de l'autre.

Le couple de problèmes de placement que l'on vient de choisir n'est pas le seul pour lequel cette affirmation est correcte. Plus généralement, il existe un lien entre certains PP et ce lien est utilisable pour la résolution d'un PP connaissant la solution de l'autre. Ceci est énoncé en détail ci-dessous.

Propriété A.7.1 (lien calculatoire)**.** *Soient E_1, E_2 les espaces de configurations identifiant deux* PP*, l'implication suivante est vérifiée :*

$$E_2 \subset E_1 \Rightarrow sol_{P,B}(E_2) = E_2 \cap sol_{P,B}(E_1)$$

Autrement dit, la solution d'un PP *identifié par E_1 fournit la solution à tout* PP *identifié par E_2, à condition que l'espace de configurations E_2 soit inclus dans l'espace de configurations E_1 et que l'on puisse calculer l'intersection : $E_2 \cap sol_{P,B}(E_1)$.*

Exemple 5. Reprenons l'exemple introductif plus en détail. Soient $E_{R.H.}, E_{R.}$ les espaces de configurations identifiant respectivement le PP par homothétie-rotation combinées et le PP par rotation. En reprenant les définitions 5.4.3 et 5.4.1 de ces espaces de configurations, le lien calculatoire implique que :

$$sol_{P,B}(E_{R.}) = E_{R.} \cap sol_{P,B}(E_{R.H.})$$

Ayant résolu le placement par homothétie-rotation combinées, les calculs nécessaires à la résolution du placement par rotation sont déjà faits. Il suffit d'identifier dans $sol_{P,B}(E_{R.H.})$ les configurations précises qui nous intéressent.

Exemple 6. Soient $E_{T.}$ et E_{Tx} les espaces de configurations identifiant respectivement le PP par translation et le PP par translation en x uniquement. Le deuxième espace de configurations mentionné est défini tel que :

$$E_{Tx} := \{(x, y_k)\} \times \{\lambda_k\} \times \{\theta_k\} \text{ avec } y_k, \lambda_k, \theta_k \text{ donnés et } x \in \mathbb{R}$$

En reprenant la définition 5.4.4 du premier espace de configurations mentionné, on déduit directement que $E_{Tx} \subsetneq E_{T.}$. Le lien calculatoire implique que :

$$sol_{P,B}(E_{Tx}) = E_{Tx} \cap sol_{P,B}(E_{T.})$$

Remarque. Selon le couple de PP que l'on considère, le calcul d'intersection est plus ou moins aisé. Pour le dernier exemple, il s'agirait simplement de calculer l'intersection entre une droite (la coordonnée x étant la seule donnée qui varie d'une configuration de $E_{Tx.}$ à l'autre) et un ensemble de disques. L'ensemble de disques en question représenterait les configurations non-autorisées trouvées dans $E_{T.}$.

Remarque. En théorie des ensembles, la relation « est un sous-ensemble de » appelée *inclusion* définit une relation d'ordre partiel sur l'ensemble des parties d'un ensemble. En transposant ceci à notre contexte, une relation d'ordre partiel existe entre les PP, suivant l'inclusion de leurs espaces de configuration respectifs.

Suite à cette remarque, nous introduisons les termes : PP supérieur et PP inférieur pour rendre compte de cette relation. On dira par exemple : le PP par rotation est inférieur au PP par homothétie-rotation combinées, ou encore : le PP par translation est supérieur au PP par translation en x uniquement.

En se basant sur cette relation, les PP mentionnés dans la thèse se répartissent selon la hiérarchie suivante :

Interprétation du calcul d'une région de VPUD comme PP : Au chapitre 3 nous avions rappelé la définition d'une région de Voronoï classique. Alternativement à cette définition bien connue (basée sur une inégalité entre distances), on peut proposer une définition équivalente qui est :

$$V(p_i) := \left\{x \in \mathbb{R}^2 \mid b\left(x, \|x - p_i\|\right) \cap P = \emptyset\right\}$$

Remarque. En analysant la condition d'appartenance à cet ensemble, on voit que calculer une région de Voronoï classique correspond finalement au problème de placer un disque unité, en lui appliquant une transformation très particulière combinant translation et homothétie. Le disque est en fait déplacé librement, mais sa taille varie, de sorte que le germe p_i se trouve toujours sur la frontière du-dit disque. Par conséquent, il existe un lien entre calcul d'une région de Voronoï et placement du disque unité.

Comme on l'a vu à la section 3.3.3, le diagramme de Voronoï est un cas particulier de diagramme de VPUD. On obtient un diagramme de Voronoï classique lorsque l'on se place dans le cas où l'EDS B est égal au singleton $B^{(1)} = \{b_1\}$ contenant le disque unité $b_1 = b(\mathcal{O}, 1)$, où $\mathcal{O}$ est l'origine du repère.

Par extension, la remarque que l'on vient de faire est valable pour toute union de disques : il existe un lien entre calcul de région de VPUD

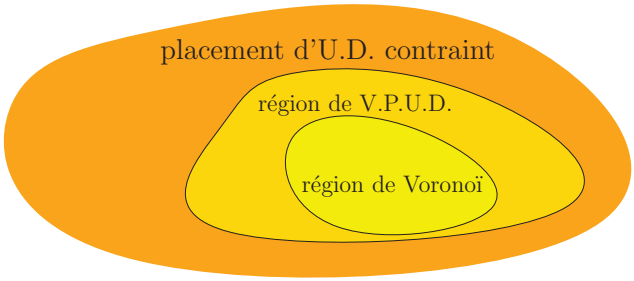


FIGURE A.4 – *Unification des concepts présentés dans le mémoire. Le calcul d'une région de Voronoï classique est un cas particulier du calcul de région de VPUD. Ce calcul lui-même est un cas particulier de résolution du placement d'union de disques dans un nuage de points*

et placement d'une *union de disques* dans un nuage de points. Ces liens conceptuels successifs sont représentés schématiquement sur la figure A.4.

Existence d'un problème de placement « spécial » : Au chapitre 5, nous avons introduit un formalisme cohérent pour désigner et étudier les différents problèmes de placement d'union de disques dans un nuage de points.

Dans la présente sous-section, en reprenant ce formalisme, nous exprimons la conjecture suivante : *il existe un* PP *dont la résolution est équivalente au calcul d'une région de VPUD*.

On désigne par « problème spécial » un tel problème de placement. Celui-ci est identifié par l'espace de configurations noté $E_{spe}(p_i)$ que nous définissons ainsi :

$$E_{spe}(p_i) := \{(x, \lambda, \theta) \in \mathbb{R}^2 \times \mathbb{R}^+ \times \{0\} \mid \lambda = \|x - p_i\|\}$$

Remarque. Cet espace de configurations possède une caractéristique non rencontrée jusqu'ici : il est paramétré par un point p_i du plan. Ce point étant choisi une fois pour toute au préalable de la résolution du problème de placement, cela ne constitue pas une difficulté nouvelle.

Tout PP spécial est identifié par un espace de configurations $E_{spe}(p_i)$. Ainsi, tout PP est défini de façon à considérer précisément les mêmes copies $\cup B(a)$ que celles qui sont considérées au cours du calcul d'une région de VPUD. La résolution directe du PP spécial semble néanmoins difficile (le PP spécial n'étant pas un PP inversible) mais cette difficulté peut être évitée comme l'explique le point suivant.

Identification d'un problème supérieur au PP spécial : Au premier paragraphe de cette section, nous avons vu qu'il existe un lien calculatoire entre PP. Cette observation est générale ; le PP spécial est donc lui aussi concerné par le lien calculatoire. Ce lien calculatoire pourrait être mis à profit pour résoudre le PP spécial, à condition de disposer d'un PP supérieur que l'on sait résoudre.

Cette partie identifie justement un tel problème de placement : il s'agit du placement par translation-homothétie combinées.

Le placement par translation-homothétie combinées cherche à placer une union de disques par translation (deux degrés de liberté) et homothétie (un degré de liberté) dans un nuage de points, afin que l'union de

disques soit vide. Ce PP est identifié par un espace de configurations que nous notons $E_{T.H.}$, défini par :

$$E_{T.H.} = \mathbb{R}^2 \times \mathbb{R}^+ \times \{\theta_k\}$$

En choisissant $\theta_k = 0$ (on détermine la valeur de l'orientation constante de l'union de disques) et en suivant les définitions d'espaces de configurations, on déduit directement que : $E_{spe}(p_i) \subset E_{T.H.}$. Le PP spécial est donc inférieur au PP par translation-homothétie combinées. Le lien calculatoire vu plus tôt implique que :

$$sol_{P,B}\left(E_{spe}(p_i)\right) = E_{spe}(p_i) \cap sol_{P,B}(E_{T.H.})$$

Ainsi, le *lien calculatoire* vu précédemment est mis à profit. Pour résoudre le PP spécial, nous allons procéder en résolvant d'abord le PP par translation-homothétie combinées. Ci-dessous nous résumons le raisonnement complet expliquant comment réaliser ce calcul.

Raisonnement permettant le calcul exact d'une région de VPUD : Notre piste de recherche pour calculer de manière exacte le diagramme de VPUD exploite les quatre observations précédentes. Récapitulons ici les étapes du raisonnement :

- Pour réaliser le calcul d'une région de VPUD, il suffit de résoudre le PP spécial ;
- pour résoudre le PP spécial, il suffit de résoudre le PP par translation-homothétie combinées *i.e.* calculer l'ensemble des configurations autorisées $sol_{P,B}(E_{T.H.})$, puis de calculer l'intersection de cet ensemble $sol_{P,B}(E_{T.H.})$ avec l'espace de configurations E_{spe}. En effet, puisque $E_{spe}(p_i) \subset E_{T.H.}$, le lien calculatoire (propriété A.7.1) nous dit que :

$$sol_{P,B}\left(E_{spe}(p_i)\right) = E_{spe}(p_i) \cap sol_{P,B}(E_{T.H.})$$

- $E_{spe}(p_i)$ peut être interprété géométriquement comme une cône droit en trois dimensions. La droite qui traverse le cône droit en son milieu est orthogonale au plan de coordonnée $z = 0$. On peut écrire :

$$E_{spe}(p_i) = cone_{droit}(p_i) := \{(x, y, z) \in \mathbb{R}^3 \mid z = \|(x, y) - p_i\|\}$$

  Or comme l'ensemble des solutions au PP par translation-homothétie combinées, noté $sol_{P,B}(E_{T.H.})$, est un sous-ensemble de $\mathbb{R}^3$ on sait calculer l'intersection entre $E_{spe}(p_i)$ et $sol_{P,B}(E_{T.H.})$;
- il suffit de résoudre le PP par translation-homothétie combinées pour résoudre le PP spécial.

Finalement, par transitivité de l'implication logique, nous concluons : *il suffit de résoudre le* PP *par translation-homothétie combinées pour calculer toute région de VPUD*. Le PP par translation-homothétie combinées n'est pas excessivement complexe. En laissant de côté la démonstration, on conjecture que l'ensemble des solutions pour ce PP, noté $sol_{P,B}(E_{T.H.})$, consiste en l'espace $\mathbb{R}^3$ privé d'un ensemble de cônes. Nous donnons quelques schémas venant appuyer cette idée intuitive.

L'idée pour obtenir ces schémas est de calculer dans un premier temps les solutions pour le PP par translation uniquement, en résolvant le PP pour

différentes valeurs constantes de λ_k. Ceci est illustré sur les figures A.5 et A.7, pour le placement d'un simple disque et pour le placement d'une union de trois disques, respectivement.

Dans un deuxième temps, en « superposant les couches », on obtient la solution pour le PP par translation-homothétie combinées. Ceci est illustré sur les figures A.6 et A.8, pour le placement d'un simple disque et pour le placement d'une union de trois disques, respectivement.

Ainsi, calculer les solutions au PP spécial est équivalent à calculer l'intersection entre un cône droit provenant de l'espace de configurations $E_{spe}(p_i)$, et plusieurs autres cônes provenant de l'ensemble des solutions $E_{T.H.}$. Ce principe est illustré par la figure A.9, cette dernière ne représente que deux cônes pour préserver la lisibilité de l'image, mais en pratique ce principe s'applique pour m cônes.

Remarque. Comme les cônes qui « creusent » l'espace $\mathbb{R}^3$ pour donner l'ensemble $sol_{P,B}(E_{T.H.})$ sont des ouverts, la surface $E_{spe}(p_i)$ intersecte l'ensemble $sol_{P,B}(E_{T.H.})$ en sa limite. Cette surface $E_{spe}(p_i)$ et les cônes sont quasi-confondus ce qui fait qu'il n'est pas aisé de voir l'interprétation géométrique du calcul. La surface $E_{spe}(p_i)$ correspond en fait au bord de l'un des cônes qui « creuse », le cône associé au premier disque de B. Remarquons aussi que l'ensemble résultant de l'intersection $E_{spe}(p_i) \cap sol_{P,B}(E_{T.H.})$ n'est pas nécessairement borné.

Une fois le PP spécial résolu, il reste une dernière étape pour passer de l'ensemble de solutions $sol_{P,B}(E_{T.H.})$ à une région de VPUD correspondante. Une région de VPUD est en effet une représentation différente des solutions au PP spécial. Notons $H(x, y, 0)$ le plan situé dans $\mathbb{R}^3$ tel que chacun de ses points a pour coordonnée $z = 0$. Il suffit de faire le projeté orthogonal des solutions du problème spécial sur le plan $H(x, y, 0)$ pour aboutir à une région plane respectant la définition de région de VPUD (définition 3.3.3). Formellement :

$$V_B(p_i) = proj\left[sol_{P,B}\left(E_{spe}(p_i)\right); H(x, y, 0)\right]$$

Pour résumer cette section en intégralité, nous proposons une piste de recherche pour le calcul exact d'une région de VPUD. Celle-ci prend la forme d'une conjecture que l'on peut écrire formellement ainsi :

$$V_B(p_i) = proj\left[E_{spe}(p_i) \cap sol_{P,B}\left(E_{T.H.}\right); H(x, y, 0)\right]$$

Cette conjecture réunit tous les concepts importants de nos travaux, elle dit que toute région de VPUD est le projeté orthogonal d'une intersection entre cônes qui modélisent respectivement : un espace de configurations, un ensemble-solution.

Ce procédé est illustré graphiquement sur la figure A.10. Sur cette même figure représentant l'intersection entre deux cônes on retrouve, après projection des configurations autorisées membres de $sol_{P,B}(E_{spe}(p_i))$ sur le plan $H(x, y, 0)$, une frontière de sur-région élémentaire. Nous avions vu, au chapitre 4, que ces régions planes (*cf.* définition 4.1.3) composent par intersection une région de VPUD complète.

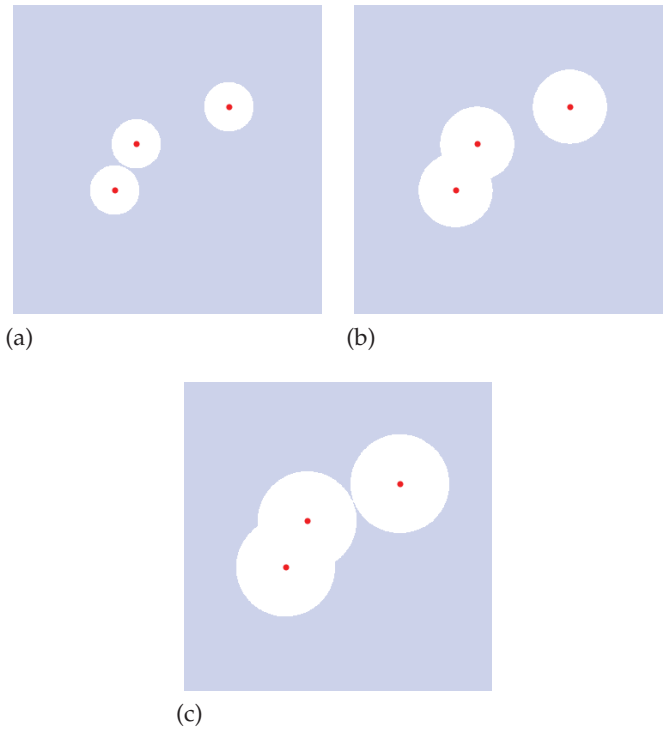


FIGURE A.5 – *Solutions au* PP *par translation d'un disque dans un nuage de trois points. L'ensemble des emplacements autorisés pour le centre du disque est représenté par la région bleue. Chacune des régions (a,b,c) est obtenue pour une taille de disque λ_k différente*

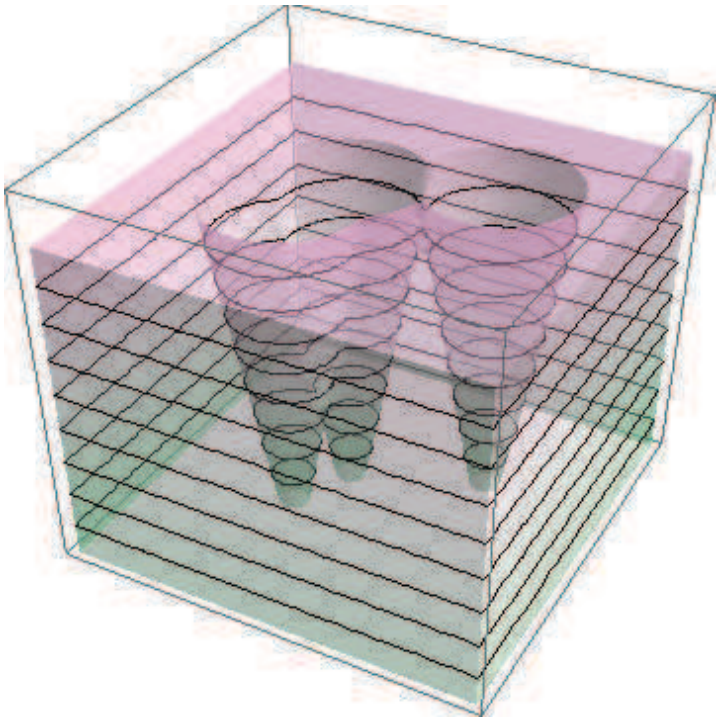


FIGURE A.6 – *Solutions au* PP *par translation-homothétie combinées d'un disque dans un nuage de trois points. L'ensemble des configurations autorisées est représenté en couleur, il constitue un sous-ensemble de $\mathbb{R}^3$ et peut être conçu comme la superposition de toutes les régions analogues à celles de la figure précédente*

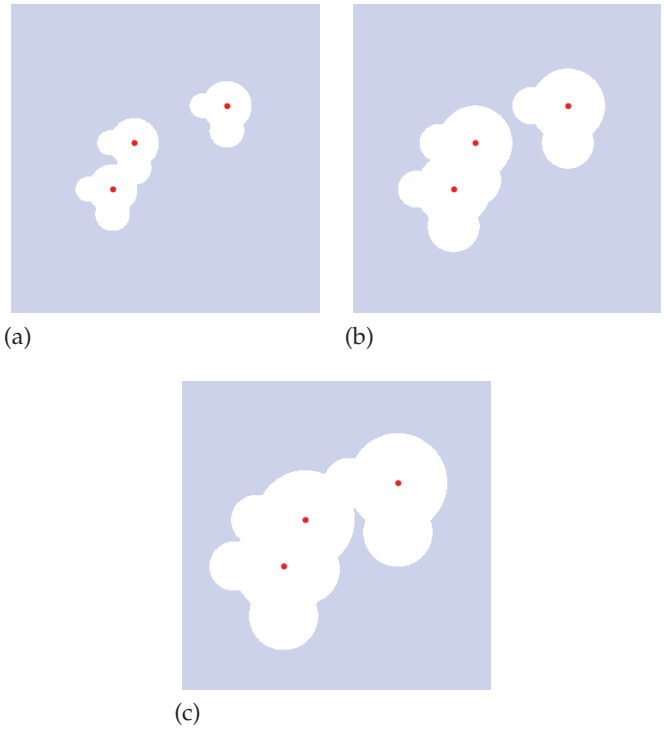


FIGURE A.7 – *Solutions au* PP *par translation d'une union de trois disques dans un nuage de trois points. L'ensemble des emplacements autorisés pour le point de référence est représenté par la région bleue. Chacune des régions (a,b,c) est obtenue pour une taille de copie* λ_k *différente*

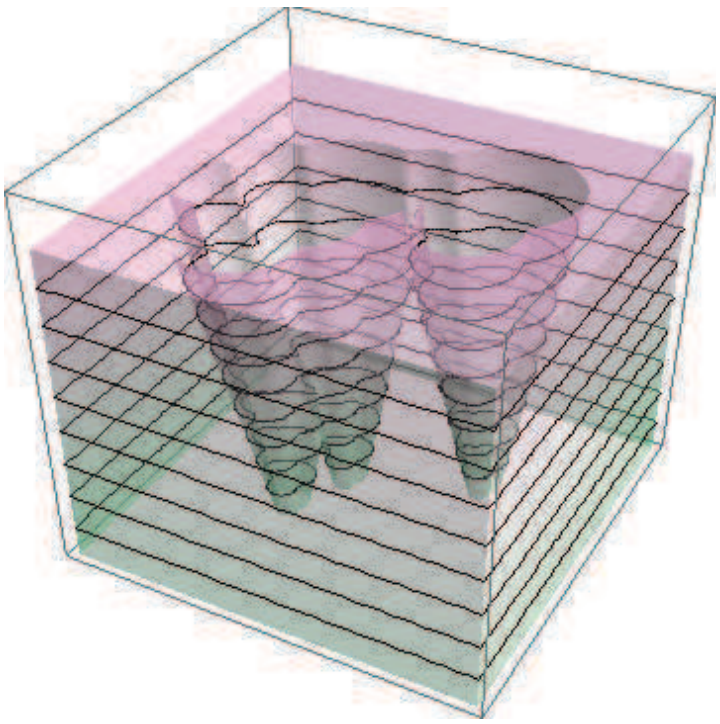
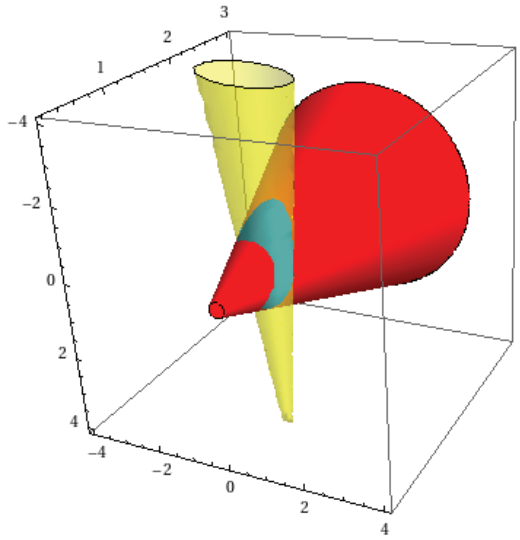
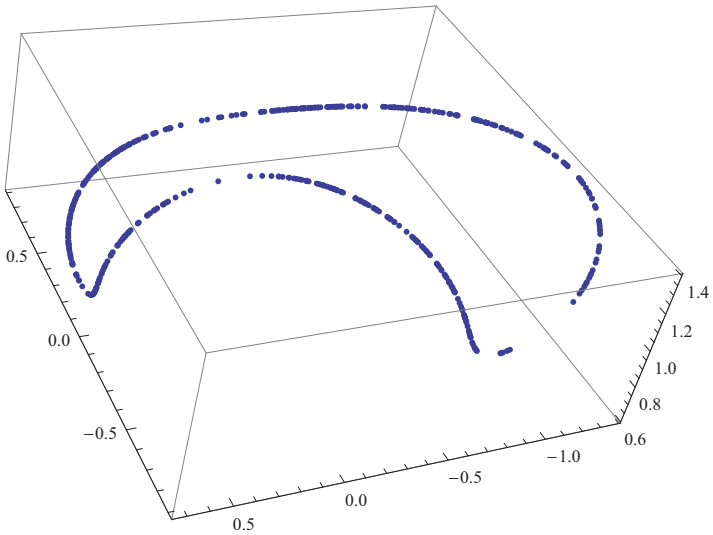


FIGURE A.8 – *Solutions au* PP *par translation-homothétie combinées d'une union de trois disques dans un nuage de trois points. L'ensemble des configurations autorisées est représenté en couleur, il constitue un sous-ensemble de* $\mathbb{R}^3$ *et peut être conçu comme une superposition de toutes les régions analogues à celles de la figure précédente*

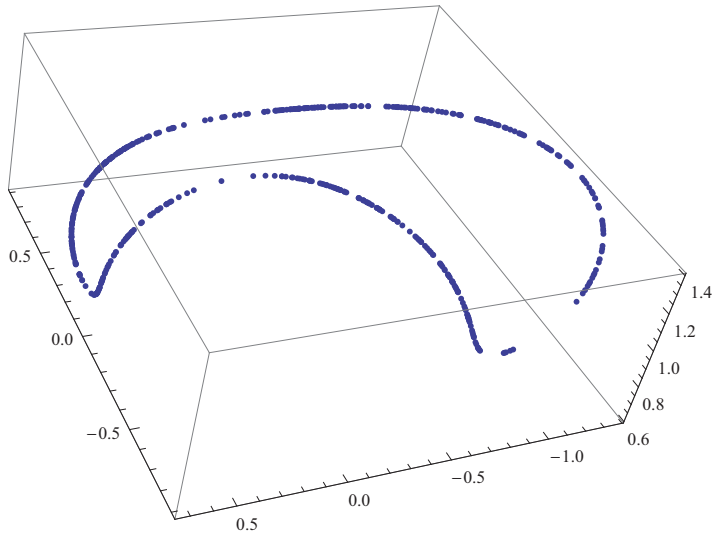


(a) *Intersection de cônes dans un espace 3D. Le cône représenté en rouge provient de l'espace de configurations $E_{spe}(p_i)$ tandis que le cône jaune provient de l'ensemble des solutions au* PP *par translation-homothétie combinées, ensemble noté $E_{T.H.}$*

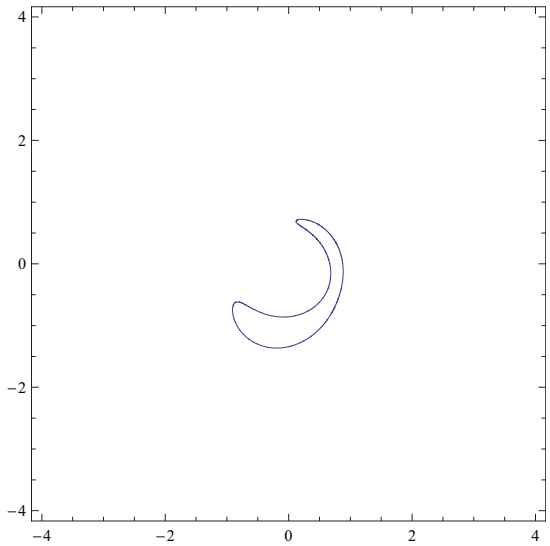


(b) *Configurations autorisées pour le* PP *spécial*

FIGURE A.9 – *Obtention des solutions au* PP *spécial en mettant à profit le lien calculatoire*



(a) *Configurations autorisées pour le* PP *spécial représentées dans* $\mathbb{R}^3$



(b) *Résultat du projeté orthogonal des configurations autorisées pour le* PP *spécial*

FIGURE A.10 – *Projection orthogonale des solution au* PP *spécial sur le plan de coordonnée* $z = 0$, *noté* $H(x, y, 0)$. *Le résultat est égal à la frontière d'une sur-région élémentaire, région définie et étudiée au chapitre 4*

Les constantes que nous avons employées pour produire cet exemple, constantes liées à P, B et à la sur-région $S_{b_k}(p_i, p_j)$ considérée, sont les suivantes :

$$\begin{aligned} p_k &= (0;0) \\ p_j &= (2;3,2) \\ c_i &= (2;4) \\ r_i &= 0.8 \end{aligned}$$

Pour conclure, les idées présentées de manière informelle dans cette section sont susceptibles d'être retranscrites algorithmiquement et programmées sur ordinateur. Cela offrirait une nouvelle méthode pour représenter sur machine et pour calculer les régions de VPUD de manière exacte.

Bibliographie

P. K. Agarwal, N. Amenta, B. Aronov, et M. Sharir. Largest placements and motion planning of a convex polygon. Dans J.-P. Laumond et M. Overmars, éditeurs, *Robotics : The Algorithmic Perspective*. A.K. Peters, Wellesley, MA, 1996. (Cité page 97.)

Pankaj K. Agarwal, Nina Amenta, et Micha Sharir. Largest placement of one convex polygon inside another. *Discrete & Computational Geometry*, 19(1) :95–104, 1998. (Cité page 101.)

Pankaj K. Agarwal, Boris Aronov, et Micha Sharir. Motion planning for a convex polygon in a polygonal environment. *Geom*, 22 :201–221, 1997. (Cité pages 14, 95, 97, 98 et 101.)

Pankaj K. Agarwal et Micha Sharir. Efficient algorithms for geometric optimization. *ACM Comput. Surv.*, 30 :412–458, December 1998. ISSN 0360-0300. (Cité page 102.)

Eugene L. Allgower, Kurt Georg, et Rick Miranda. The method of resultants for computing real solutions of polynomial systems. *SIAM J. Numer. Anal.*, 29(3) :831–844, Juin 1992. ISSN 0036-1429. URL `http://dx.doi.org/10.1137/0729051`. (Cité page 67.)

Nina Amenta, Sunghee Choi, et Ravi Krishna Kolluri. The power crust. Dans *Proceedings of the sixth ACM symposium on Solid modeling and applications*, SMA '01, pages 249–266, New York, NY, USA, 2001. ACM. ISBN 1-58113-366-9. URL `http://doi.acm.org/10.1145/376957.376986`. (Cité pages 3 et 23.)

John Augustine, Brian Putnam, et Sasanka Roy. Largest empty circle centered on a query line. *J. of Discrete Algorithms*, 8(2) :143–153, Juin 2010. ISSN 1570-8667. URL `http://dx.doi.org/10.1016/j.jda.2009.10.002`. (Cité page 98.)

F Aurenhammer. Power diagrams : properties, algorithms and applications. *SIAM J. Comput.*, 16(1) :78–96, Février 1987. ISSN 0097-5397. URL `http://dx.doi.org/10.1137/0216006`. (Cité page 37.)

Franz Aurenhammer. Voronoi diagrams –a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23, 1991. (Cité pages 5 et 36.)

Franz Aurenhammer et Rolf Klein. Voronoi diagrams. Dans J.-R. Sack et J. Urrutia, éditeurs, *Ch. 5 in Handbook of Computational Geometry*, pages 201–290, 2000. (Cité pages 7, 35 et 36.)

Francis Avnaim et Jean-Daniel Boissonnat. Polygon placement under translation and rotation. Dans *Proceedings of the 5th Annual Symposium on Theoretical Aspects of Computer Science*, STACS '88, pages 322–333, London, UK, UK, 1988. Springer-Verlag. ISBN 3-540-18834-7. URL http://dl.acm.org/citation.cfm?id=646504.696465. (Cité pages 97, 98 et 101.)

Harry Blum. A Transformation for Extracting New Descriptors of Shape. Dans Weiant Wathen-Dunn, éditeur, *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, Cambridge, 1967. (Cité page 3.)

Christoph Burnikel, Kurt Mehlhorn, et Stefan Schirra. How to compute the voronoi diagram of line segments : Theoretical and experimental results. Dans *ESA*, pages 227–239, 1994. (Cité page 37.)

John Canny. Piano movers problems. Dans Oussama Khatib, John J. Craig, et Tomás Lozano-Pérez, éditeurs, *The robotics review 1*, Chapitre On the Piano Movers series by Schwartz, Sharir, and Ariel-Sheffi, pages 33–40. MIT Press, Cambridge, MA, USA, 1989. (Cité page 14.)

Bernard Chazelle. The polygon containment problem. Dans *Advances in Computing Research*, 1983. (Cité pages 14, 93, 97, 98 et 101.)

Bernard Chazelle, Robert L. (Scot) Drysdale III, et D. T. Lee. Computing the largest empty rectangle. *SIAM J. Comput.*, 15(1) :300–315, 1986. (Cité pages 93, 97 et 102.)

Mao Chen et Wenqi Huang. Heuristic algorithm for packing triangles into a square container. *International journal of information and management sciences*, 20(2) :255–268, 2009. ISSN 1017-1819. (Cité page 103.)

L. P. Chew et K. Kedem. Placing the largest similar copy of a convex polygon among polygonal obstacles. Dans *Proceedings of the fifth annual symposium on Computational geometry*, SCG '89, pages 167–173, New York, NY, USA, 1989. ACM. ISBN 0-89791-318-3. URL http://doi.acm.org/10.1145/73833.73853. (Cité pages 94, 97, 98 et 102.)

L. P. Chew et K. Kedem. A convex polygon among polygonal obstacles : Placement and high-clearance motion. *Computational Geometry : Theory and Applications*, 3 :59–89, 1993. (Cité page 14.)

L. Paul Chew et Robert L. Drysdale. Finding Largest Empty Circles with Location Constraints. Rapport Technique PCS-TR86-130, Dartmouth College, Computer Science, Hanover, NH, 1986. URL http://www.cs.dartmouth.edu/reports/TR86-130.pdf. (Cité pages 97, 98 et 102.)

L.P. Chew et R.L. Drysdale III. Voronoi diagrams based on convex distance functions. *Proceedings of the first annual symposium on Computational geometry*, pages 235–244, 1985. (Cité page 37.)

George E. Collins. The calculation of multivariate polynomial resultants. *J. ACM*, 18(4) :515–532, Octobre 1971. ISSN 0004-5411. URL http://doi.acm.org/10.1145/321662.321666. (Cité page 67.)

José Miguel Díaz-Báñez, Ferran Hurtado, Henk Meijer, David Rappaport, et Joan Antoni Sellarès. The largest empty annulus problem. *Int. J. Comput. Geometry Appl.*, 13(4) :317–325, 2003. (Cité pages 97, 98 et 102.)

Rex A. Dwyer et William F. Eddy. Maximal empty ellipsoids. Dans *Proceedings of the fifth annual ACM-SIAM symposium on Discrete algorithms*, SODA '94, pages 98–102, Philadelphia, PA, USA, 1994. Society for Industrial and Applied Mathematics. ISBN 0-89871-329-3. URL http://dl.acm.org/citation.cfm?id=314464.314483. (Cité page 98.)

C. Ericson. *Real-Time Collision Detection*. Numéro vol. 1 dans Morgan Kaufmann Series in Interactive 3D Technology. Elsevier, 2005a. ISBN 9781558607323. URL http://books.google.fr/books?id=WGpL6Sk9qNAC. (Cité page 95.)

Christer Ericson. *Real-Time Collision Detection (The Morgan Kaufmann Series in Interactive 3-D Technology)*. Morgan Kaufmann, Janvier 2005b. ISBN 1558607323. (Cité page 14.)

S. Fortune. A fast algorithm for polygon containment by translation. Dans Wilfried Brauer, éditeur, *Automata, Languages and Programming*, volume 194 de *Lecture Notes in Computer Science*, pages 189–198. Springer Berlin / Heidelberg, 1985. ISBN 978-3-540-15650-5. URL http://dx.doi.org/10.1007/BFb0015744. 10.1007/BFb0015744. (Cité pages 97 et 101.)

S. Goedecker. Remark on algorithms to find roots of polynomials. *SIAM J. Sci. Comput.*, 15(5) :1059–1063, Septembre 1994. ISSN 1064-8275. URL http://dx.doi.org/10.1137/0915064. (Cité page 67.)

R.K. Guy, H.T. Croft, et K.J. Falconer. *Unsolved Problems in Geometry*. Problem books in mathematics. Springer-Verlag, 1994. ISBN 9780387975061. URL http://books.google.fr/books?id=8zjBjwEACAAJ. (Cité page 102.)

D. Halperin et M. Sharir. A near-quadratic algorithm for planning the motion of a polygon in a polygonal environment. *Discrete and Computational Geometry*, 16 :121–134, 1996. (Cité pages 97 et 101.)

Shai Hirsch et Dan Halperin. Hybrid motion planning : Coordinating two discs moving among polygonal obstacles in the plane. Dans *Algorithmic Foundations of Robotics V*, pages 239–255. Springer, 2002. (Cité pages 14 et 95.)

Philip M. Hubbard. Approximating polyhedra with spheres for time-critical collision detection. *ACM Trans. Graph.*, pages 179–210, 1996. (Cité page 21.)

D. Husemöller. *Elliptic curves*. Graduate texts in mathematics. Springer-Verlag, 1987. ISBN 9783540963714. URL http://books.google.com.na/books?id=T-2_QgAACAAJ. (Cité page 55.)

Thomas Iwaszko, Mahmoud Melkemi, et Lhassane Idoumghar. A theoretical structure for computational geometry : regions of point-free overlapping circles. Dans *Proceedings of the 9th WSEAS international conference on signal processing, computational geometry and artificial vision*, ISCGAV'09, pages 117–122, 2009a. ISBN 978-960-474-108-3. (Cité page 39.)

Thomas Iwaszko, Mahmoud Melkemi, et Lhassane Idoumghar. A theoretical structure for computational geometry : regions of point-free overlapping circles. Dans *Abstracts from the 7th Japan Conference on Computational Geometry and Graphs*, pages 81—82, 2009b. (Cité page 39.)

Thomas Iwaszko, Mahmoud Melkemi, et Lhassane Idoumghar. A geometric data structure applicable to image mining and retrieval. Dans *Proceedings of the 7th International Conference on Image Analysis and Recognition (1)*, pages 364–373, 2010. (Cité page 39.)

S. Jakobs. On genetic algorithms for the packing of polygons. *European Journal of Operational Research*, 88(1) :165–181, Janvier 1996. ISSN 03772217. URL `http://dx.doi.org/10.1016/0377-2217(94)00166-9`. (Cité page 103.)

L. Jaulin, M. Kieffer, O. Didrit, et E. Walter. *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics*. Springer-Verlag, London, 2001. (Cité page 79.)

Henri Fraysse Jean-Marie Arnaudiès. *Algèbre bilinéaire et géométrie, vol 4*. Dunod, 1996. (Cité page 56.)

Menelaos I. Karavelas et Mariette Yvinec. Dynamic additively weighted voronoi diagrams in 2d. Dans *Proceedings of the 10th Annual European Symposium on Algorithms*, ESA '02, pages 586–598, London, UK, UK, 2002. Springer-Verlag. ISBN 3-540-44180-8. URL `http://dl.acm.org/citation.cfm?id=647912.740653`. (Cité page 36.)

Menelaos I. Karavelas et Mariette Yvinec. The voronoi diagram of planar convex objects. Dans *In European Symp.on Algorithms*, pages 337–348, 2003. (Cité page 37.)

Robert A. Katz et Stephen M. Pizer. Untangling the blum medial axis transform. *International Journal of Computer Vision*, 55 :139–153, 2001. (Cité page 3.)

R. Klein. Abstract voronoi diagrams and their applications (extended abstract). Dans *Proceedings on International Workshop on Computational Geometry and its Applications*, pages 148–157, New York, NY, USA, 1988. Springer-Verlag New York, Inc. ISBN 0-387-50335-8. (Cité pages 36 et 37.)

Alasdair T. Laurie et Richard M. Jackson. Methods for the prediction of protein-ligand binding sites for structure-based drug design and virtual ligand screening. *Current protein & peptide science*, 7(5) :395–406, Octobre 2006. (Cité page 14.)

D. T. Lee. Two-dimensional voronoi diagrams in the lp-metric. *J. ACM*, 27 (4) :604–618, 1980. ISSN 0004-5411. (Cité page 36.)

D. T. Lee et R. L. Drysdale. Generalization of Voronoi diagrams in the plane. *SIAM Journal on Computing*, 10 :73–87, 1981. (Cité page 36.)

D. T. Lee et C. K. Wong. Voronoi diagrams in l_1 (l_infty) metrics with 2-dimensional storage applications. *SIAM J. Comput.*, pages 200–211, 1980. (Cité page 7.)

D. Leven et M. Sharir. Planning a purely translational motion for a convex object in two-dimensional space using generalized Voronoi diagrams. *Discrete and Computational Geometry*, 2 :9–31, 1987a. (Cité page 101.)

D. Leven et M. Sharir. Planning a purely translational motion for convex objects in two dimensional space using generalized voronoi diagrams. *Discrete & Computational Geometry*, 2 :9–31, 1987b. (Cité pages 14 et 97.)

Zhenyu Li et Victor Milenkovic. Compaction and separation algorithms for non-convex polygons and their applications. *European Journal of Operations Research*, 84 :539–561, 1995. (Cité page 103.)

Sebastien Loriot. *Arrangements of circles on a sphere : Algorithms and applications to molecular models represented by a union of balls*. PhD thesis, Université de Bourgogne, 2008. (Cité page 14.)

Serge Mehl. Courbes algébriques planes, 2009 (accédé le 5 mars 2012). http ://serge.mehl.free.fr/anx/courbe_alg.html. (Cité page 59.)

Dominique Michelucci et Sebti Foufou. Interval-based tracing of strange attractors. *International Journal of Computational Geometry and Applications*, 16(1) :27–40, 2006. (Cité page 79.)

D. P. Mitchell. Robust ray intersection with interval arithmetic. Dans *Proceedings on Graphics interface '90*, pages 68–74, Toronto, Ont., Canada, Canada, 1990. Canadian Information Processing Society. (Cité page 79.)

Don Mitchell. Three applications of interval analysis in computer graphics. Dans *SIGGRAPH,Course Notes*, 1991. (Cité page 79.)

R. E. Moore. *Interval Arithmetic and Automatic Error Analysis in Digital Computing*. Ph.D. dissertation, Department of Mathematics, Stanford University, Stanford, CA, USA, Novembre 1962. URL `http://interval.louisiana.edu/Moores_early_papers/disert.pdf`. Also published as Applied Mathematics and Statistics Laboratories Technical Report No. 25. (Cité page 79.)

A Naamad, D.T Lee, et W.-L Hsu. On the maximum empty rectangle problem. *Discrete Applied Mathematics*, 8(3) :267 – 277, 1984. ISSN 0166-218X. URL `http://www.sciencedirect.com/science/article/pii/0166218X84901240`. (Cité page 97.)

Pierre Nicaise. *Les Courbes algébriques planes du troisième ordre*. Publibook, 2011. (Cité page 56.)

Frank Nielsen et Richard Nock. Skew jensen-bregman voronoi diagrams. *Transactions on Computational Science*, 14 :102–128, 2011. (Cité page 37.)

Atsuyuki Okabe, Barry Boots, Kokichi Sugihara, et Sung Nok Chiu. *Spatial Tessellations : Concepts and Applications of Voronoi Diagrams*. Series in Probability and Statistics. John Wiley and Sons, Inc., 2nd ed. édition, 2000. (Cité pages 5 et 35.)

Étienne Parizot et Jean-Daniel Boissonnat. Manipulation informatique des objets géométriques, 2004. `http://interstices.info/objets-geometriques` (accédé le 8 mars 2012). (Cité page 93.)

Helmut Ratschek et Jon Rokne. *New computer methods for global optimization*. Halsted Press, New York, NY, USA, 1988. ISBN 0-470-21208-X. (Cité page 79.)

Natalie Révol. Introduction à l'arithmétique des intervalles. *École des jeunes chercheurs*, 2004. (Cité page 79.)

Micha Sharir. Efficient algorithms for planning purely translational collision-free motion in two and three dimensions. Dans *Proc. of the IEEE Int. Conf. on Robotics and Automation, Raleigh NC*, pages 1326–1331, 1987. (Cité pages 97 et 101.)

Micha Sharir et Pankaj K. Agarwal. *Davenport-Schinzel Sequences and their Geometric Applications*. Cambridge University Press, New York, NY, USA, 2010. ISBN 0521135117, 9780521135115. (Cité page 15.)

Micha Sharir et Sivan Toledo. Extremal polygon containment problems. *Computational Geometry Theory and Application*, 4 :99–118, June 1994. ISSN 0925-7721. (Cité pages 97, 98 et 102.)

Kenneth Stephenson. Circle packing : a mathematical tale. *Notices Amer. Math. Soc*, 50 :1376–1388, 2003. (Cité pages 21 et 102.)

Roger Tam et Alain Fournier. Shape model and threshold extraction via shape gradients. Dans *In Proc. Vision, Modeling and Visualization*, pages 481–489, 2001. (Cité page 23.)

Roger Tam et Wolfgang Heidrich. Shape simplification based on the medial axis transform. Dans *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, VIS '03, pages 63–, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 0-7695-2030-8. URL `http://dx.doi.org/10.1109/VISUAL.2003.1250410`. (Cité page 3.)

Roger C. Tam. *Voronoi ball models for computational shape applications*. PhD thesis, The University of British Columbia (Canada), 2004. AAINQ90276. (Cité pages ix, 3, 21 et 23.)

Godfried T. Toussaint. Computing largest empty circles with location constraints. *International Journal of Computer and Information Sciences*, 12 (5) :347–358, 1983. (Cité pages 36, 97, 98 et 102.)

Georgy Voronoï. Nouvelles applications des parametres continus a la theorie des formes quadratiques. *Journal für die reine und angewandte Mathematik*, 1(134) :198–287, 1908. (Cité page 36.)

Notations

$\mathbb{R}, \mathbb{R}^+$	ensembles des nombres réels, nombres réels positifs respectivement
P	nuage de points *i.e.* ensemble fini de points du plan euclidien
B	ensemble de disques standard (qui respecte la définition 2.2.1)
b_i	disque, aussi noté $b(c_i, r_i)$, élément de l'ensemble B
$\cup B$	région du plan égale à l'union de tous les disques de B
a	configuration, aussi notée (u, λ, θ), élément de Ω
Ω	ensemble de toutes les configurations, $\Omega = \mathbb{R}^2 \times \mathbb{R}^+ \times [0, 2\pi[$
$\cup B(a)$	région similaire à $\cup B$, appelée copie, identifiée par une configuration a
$b_i(a)$	disque, aussi noté $b(c_i(a), r_i(a))$, élément de l'ensemble $B(a)$
$c_i(a)$	centre du i-ème disque de la copie $\cup B(a)$
c_i	fonction-centre *i.e.* fonction associant à toute configuration a le point $c_i(a)$
VPUD	abréviation pour : Voronoï Parametrée(s) par une Union de Disques
E	désigne l'espace de configurations d'un problème de placement quelconque
PP	abréviation pour : Problème de Placement
$E_{R.}$	espace de configurations identifiant le PP par rotation
$E_{H.}$	espace de configurations identifiant le PP par homothétie
$E_{HR.}$	espace de configurations identifiant le PP par homothétie-rotation combinée
$E_{T.}$	espace de configurations identifiant le PP par translation

Ce document a été préparé à l'aide de l'éditeur de texte gedit et du logiciel de composition typographique LaTeX 2ε.

Titre généralisation du diagramme de Voronoï et placement de formes géométriques complexes dans un nuage de points

Résumé La géométrie algorithmique est une discipline en pleine expansion dont l'objet est la conception d'algorithmes résolvant des problèmes géométriques. Ceux-ci sont très utiles notamment dans l'ingénierie, l'industrie, et le multimédia. Pour être efficaces, ils utilisent fréquemment des structures de données spécifiques. Dans cette thèse nous nous sommes intéressé à une telle structure : le diagramme de Voronoï, en proposant une généralisation de celui-ci. Nous avons étendu le prédicat du disque propre à toute région de Voronoï à une union de disques. Nous avons étudié les régions basées sur ce nouveau prédicat et avons conçu des méthodes pour les calculer. Par ailleurs, Nous nous sommes aussi intéressé aux « problèmes de placement » qui ont été souvent étudiés en géométrie algorithmique. Nous avons conçu un cadre global pour étudier de tels problèmes et avons proposé une méthode générique apte à résoudre pour la première fois plusieurs problèmes de placement différents. Nos travaux ont comme intérêt d'étendre le champ d'application de ces outils, tout en unifiant leurs définitions.

Mots-clés diagramme de Voronoï généralisé, placement de formes géométriques, planification de trajectoire, vision par ordinateur

Title generalizing the Voronoi diagram and placing complex geometric shape among a point-set

Abstract Computational geometry is an active branch of computer science whose goal is the design of efficient algorithms solving geometric problems. Such algorithms are useful in domains like engineering, industry and multimedia. In order to be efficient, algorithms may use special data structure. In this thesis we focused on such a structure : the Voronoi diagram and proposed a new generalized diagram. We have extended the empty disk predicate to arbitrary union of disks. We have studied the resulting plane regions based on the new predicate, and designed methods for computing them. Then we have considered a second topic, namely the shape placement problems. Those problems are often studied in computational geometry. We designed a global framework for studying such problem, and proposed a generic method which, for the first time, is able to solve different placement problems. Thus, our work extend the scope of these tools and aims at unifying their definitions.

Keywords generalized Voronoi diagram, shape placement, motion planning, machine vision