

Numéro d'ordre : 432

THÈSE

présentée à

l'Université Strasbourg – École doctorale MSII  
Laboratoire des Sciences de l'Image, de l'Informatique et de la Télédétection  
UMR 7005 CNRS/ULP

par

**Mlle Claire BAEGERT**

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE STRASBOURG  
DISCIPLINE : INFORMATIQUE

---

**PLANIFICATION ASSISTÉE PAR ORDINATEUR POUR LA  
THERMOABLATION DE TUMEUR**

---

soutenue publiquement le 11 décembre 2009  
devant le jury composé de :

M. Stéphane COTIN, rapporteur externe  
M. Marc DANIEL, rapporteur externe  
M. Michel DE MATHELIN, examinateur  
M. Pascal SCHRECK, directeur de thèse  
Mme Caroline ESSERT-VILLARD, invitée  
M. Luc SOLER, invité



## Remerciements

EN premier lieu, je tiens à remercier mon directeur de thèse Pascal Schreck ainsi que Caroline Essert. Pascal, ton influence sur mes travaux est indéniable et tu as toujours su mettre le doigt sur les aspects les plus confus de mon travail, me poussant à les éclaircir ou les étoffer. Caroline tu as toujours été très présente et tu m’as donné régulièrement des objectifs concrets qui m’ont certainement permis d’avancer plus efficacement. Vos qualités en tant qu’encadrants se complètent. J’ai beaucoup apprécié la liberté et la confiance que vous m’avez accordées, tout en étant disponibles quand j’en avais besoin.

PAR ailleurs, je remercie Marc Daniel et Stéphane Cotin, tous deux rapporteurs de cette thèse, pour l’intérêt qu’ils ont porté à mon manuscrit. Leurs nombreuses remarques ont été très instructives et j’ai beaucoup apprécié leur vision externe de mes travaux. Merci également Michel de Mathelin pour avoir accepté de présider mon jury de thèse.

J’adresse ma reconnaissance à la région Alsace ainsi que l’IRCAD qui ont financé conjointement cette thèse. Merci à Luc Soler pour m’avoir accueillie dans son équipe de recherche. Je tiens également à remercier tous les membres de l’équipe, pour l’accueil chaleureux qu’ils m’ont réservé, l’ambiance de travail motivante et les pauses et repas partagés. En particulier je souhaite remercier Nicolas et Guillaume qui ont eu l’amabilité de répondre à mes questions techniques. Merci à Anne-Blandine qui a eu la gentillesse de reconstruire les modèles de patients sur lesquels j’ai travaillé. Je tiens aussi à remercier le Pr Gangi et le Dr Buy, pour m’avoir accordé un peu de leur temps précieux à chaque fois que j’ai assisté à l’une de leurs opérations. Je n’oublie pas l’équipe IGG dans son ensemble qui m’a toujours considérée comme membre à part entière bien que peu présente dans les locaux.

JE pense évidemment à tous mes compagnons thésards, ou devrais-je dire docteurs, qui m’ont accompagnée sur un bout du chemin : Alex, Arnaud, Ben, Guillaume, Seb. Votre présence et votre succès m’ont motivé à persévérer. Je n’oublie pas tous mes amis que j’ai connu pendant ma période universitaire et qui ont contribué à faire de celle-ci une période à laquelle je penserai probablement avec nostalgie. Un grand merci également à mes amies qui m’ont changé les idées et apporté de la bonne humeur.

JE souhaite également remercier mes parents qui ont pris le temps de relire mon manus-

crit. Mais surtout je les remercie ainsi que toute ma famille et belle-famille dans son ensemble pour m'avoir toujours soutenue lors de ces longues études, mais aussi avant, et j'en suis sûre après.

COMME, j'aime garder le meilleur pour la fin, je souhaite enfin remercier les deux hommes de ma vie. Nicolas, merci de m'avoir accompagné tout ce temps, malgré le stress, les découragements et, parfois, la mauvaise humeur. Plus spécifiquement merci d'avoir supporté mes multiples répétitions de présentation, mais également pour la préparation de mon pot de thèse qui restera, j'en suis sûre dans les mémoires gustatives de tous. Alexis, mon petit bonhomme, merci pour le temps passé avec toi. Tu éclaires mes journées et tu animes mes nuits. Ton sourire a rendu cette fin de thèse un tout petit peu plus longue mais tellement plus agréable.

# Table des matières

<b>Introduction</b>	<b>1</b>
<b>1 Contexte médical</b>	<b>5</b>
1.1 Apport de l’informatique aux opérations de chirurgie mini-invasives . . . .	6
1.1.1 Les interventions chirurgicales minimalement invasives . . . . .	6
1.1.2 Techniques d’imagerie médicale . . . . .	6
1.1.3 Visualisation 3D . . . . .	7
1.1.4 Aide à la décision . . . . .	9
1.1.5 Simulation . . . . .	10
1.1.6 Assistance per-opérateur . . . . .	10
1.2 Thérapies minimalement invasives pour l’ablation de tumeurs . . . . .	12
1.2.1 La thermo-ablation par radiofréquence . . . . .	13
1.2.1.1 Principe . . . . .	13
1.2.1.2 Indications . . . . .	14
1.2.1.3 Zone de coagulation conséquente au traitement . . . . .	15
1.2.1.4 Etape de planification : choix d’une stratégie d’intervention	16
1.2.2 Autres types d’ablation minimalement invasives . . . . .	18
1.2.2.1 Cryoablation . . . . .	18
1.2.2.2 Alcoolisation . . . . .	18
1.2.2.3 Autres opérations de thermo-ablation . . . . .	19
1.3 Planification pré-opératoire assistée par ordinateur . . . . .	20
1.3.1 Simulation du traitement . . . . .	20
1.3.2 Aide à la décision . . . . .	22
1.3.2.1 Cryochirurgie de la prostate . . . . .	22
1.3.2.2 Thermo-ablations de tumeurs hépatiques . . . . .	23
1.3.2.3 Interventions minimalement invasives robotisées . . . . .	23
1.3.2.4 Problématiques communes aux différentes interventions .	24
1.3.2.5 Positionnement de ces travaux . . . . .	24
<b>2 Contraintes et optimisation</b>	<b>27</b>
2.1 Modélisation géométrique . . . . .	29
2.1.1 Modélisation : approche déclarative <i>vs</i> approche procédurale . . . .	29
2.1.2 Modélisation géométrique en CAO . . . . .	30
2.1.3 Modélisation par contraintes . . . . .	31

---

2.1.4	La modélisation déclarative . . . . .	32
2.1.4.1	Le processus de modélisation . . . . .	32
2.1.4.2	Modeleurs déclaratifs . . . . .	34
2.1.5	L'approche déclarative pour la planification d'opération mini-invasive	35
2.2	Optimisation . . . . .	36
2.2.1	Définitions et terminologie . . . . .	36
2.2.2	Méthodes d'optimisation monocritère multidimensionnelles . . . . .	37
2.2.2.1	Les méthodes d'optimisation locale . . . . .	38
2.2.2.2	Les méta-heuristiques globales . . . . .	40
2.2.3	Choix d'une méthode d'optimisation . . . . .	42
2.3	Les problèmes d'optimisation multi-objectifs . . . . .	43
2.3.1	Problématique . . . . .	43
2.3.2	Optimalité au sens de Pareto . . . . .	43
2.3.3	Méthodes de résolution muticritères . . . . .	44
2.3.3.1	Les méthodes agrégées . . . . .	45
2.3.3.2	Méthodes traitant séparément les objectifs . . . . .	48
2.3.3.3	Les méthodes Pareto . . . . .	48
2.3.4	Discussion . . . . .	49
<b>3</b>	<b>Formalisation des connaissances</b>	<b>51</b>
3.1	Construction d'un terme représentant une contrainte . . . . .	54
3.1.1	Types manipulés . . . . .	54
3.1.2	Opérateurs . . . . .	55
3.1.3	Représentation des termes sous forme d'arbres . . . . .	56
3.1.4	Exemples de termes . . . . .	57
3.2	Connaissances à formaliser . . . . .	58
3.2.1	Distinction contraintes strictes / contraintes souples . . . . .	58
3.2.2	Choix de la nature d'une contrainte à formaliser . . . . .	59
3.2.3	L'introduction de données complémentaires . . . . .	61
3.3	Définition de contraintes grâce au formalisme XML . . . . .	62
3.3.1	Spécificité des données complémentaires . . . . .	63
3.3.2	Reformulation du problème défini par les contraintes strictes . . . . .	64
3.3.2.1	Définition de termes correspondant à des contraintes strictes	66
3.3.3	Contraintes souples et optimisation . . . . .	67
3.3.4	Conformité du fichier de contrainte . . . . .	68
3.3.4.1	Désactivation d'une contrainte dont le terme n'est pas re- connu . . . . .	68
3.3.4.2	Fichier de contraintes minimal . . . . .	69
3.4	Fichier de contraintes correspondant à la radiofréquence hépatique . . . . .	70
3.4.1	Les contraintes à formaliser . . . . .	70
3.4.2	Définition des contraintes strictes . . . . .	73

3.4.2.1	"L'aiguille traverse la peau et sa pointe doit être placée dans la tumeur traitée" . . . . .	73
3.4.2.2	"Le volume traité doit englober la totalité du volume de la tumeur accompagnée d'une marge de 5 mm." . . . . .	73
3.4.2.3	"L'aiguille ne doit pas endommager les organes vitaux du patient" . . . . .	73
3.4.2.4	"La longueur de l'aiguille doit permettre d'atteindre la tumeur " . . . . .	75
3.4.2.5	"La trajectoire de l'aiguille ne doit pas être trop tangente à la surface du foie" . . . . .	75
3.4.2.6	"La trajectoire de l'aiguille doit inclure une portion de foie sain" . . . . .	76
3.4.3	Les contraintes strictes : choix de formalisation . . . . .	76
3.4.4	Définition des contraintes souples . . . . .	77
3.4.4.1	"Le volume de tissus sains nécrosés par le traitement doit être minimal" . . . . .	77
3.4.4.2	"La trajectoire de l'aiguille doit être la plus directe possible" . . . . .	77
3.4.4.3	"Les trajectoires frôlant les organes vitaux doivent être évitées" . . . . .	78
3.4.5	L'agrégation des contraintes souples . . . . .	79
3.4.5.1	Mise à l'échelle des différentes contraintes . . . . .	79
3.4.5.2	Contrainte représentant la fonction objectif . . . . .	82
3.5	La formalisation des contraintes : conclusions et perspectives . . . . .	83
<b>4</b>	<b>Résolution des contraintes de planification</b> . . . . .	<b>85</b>
4.1	Approche objet pour l'évaluation des contraintes . . . . .	87
4.1.1	Evaluation d'un terme . . . . .	87
4.1.1.1	Processus récursif d'évaluation . . . . .	87
4.1.1.2	Noeuds à valeurs variables . . . . .	89
4.1.2	Limitations des évaluations redondantes . . . . .	89
4.2	Un opérateur spécifique pour le problème de l'évitement d'obstacles . . . . .	91
4.2.1	Le problème de la contrainte d'évitement d'obstacles . . . . .	91
4.2.2	Le traitement des obstacles grâce à un algorithme de calcul de visibilité . . . . .	92
4.2.2.1	Détermination de la visibilité depuis un point . . . . .	92
4.2.2.2	Extension de l'algorithme pour calculer la visibilité depuis un volume . . . . .	94
4.3	Le problème d'optimisation sous contraintes à résoudre . . . . .	95
4.3.1	Réduction des degrés de liberté . . . . .	95
4.3.1.1	Suppression du paramètre de puissance . . . . .	95
4.3.1.2	Fonction associant placement et puissance dans le cas de la RFA . . . . .	96

4.3.1.3	Simulation physique du volume d'effet . . . . .	96
4.3.2	Séparation du traitement des contraintes strictes et des contraintes souples . . . . .	97
4.4	Détermination de l'espace des solutions . . . . .	99
4.4.1	Les contraintes strictes d'affectation . . . . .	99
4.4.2	Les contraintes strictes restrictives . . . . .	100
4.5	Optimisation des contraintes souples . . . . .	103
4.5.1	Problématique générale . . . . .	103
4.5.2	Echantillonnage de l'espace des solutions . . . . .	104
4.5.3	Choix des échantillons d'intérêt . . . . .	105
4.6	Validation de l'heuristique sur un exemple de fonction objectif . . . . .	107
4.6.1	Algorithme quasi-exhaustif pour le calcul de l'optimum global de la fonction étudiée . . . . .	108
4.6.1.1	Elimination du problème des minima locaux en fixant une partie des paramètres . . . . .	108
4.6.1.2	Méthode de calcul du minimum théorique . . . . .	111
4.6.2	Comparaison des résultats fournis par l'heuristique proposée avec le minimum théorique . . . . .	111
4.7	Résolution des contraintes : conclusion et perspectives . . . . .	112
<b>5</b>	<b>le logiciel de planification RF-Sim</b>	<b>115</b>
5.1	Structure et implémentation du logiciel de planification . . . . .	116
5.1.1	Interprétation du fichier de contraintes . . . . .	117
5.1.2	Solveur . . . . .	119
5.1.3	Scène 3D . . . . .	120
5.1.4	Interface . . . . .	121
5.2	Fonctionnalités de l'interface utilisateur . . . . .	121
5.2.1	Visualisation simultanée 2D/3D . . . . .	121
5.2.2	Matérialisation de l'espace des solutions . . . . .	123
5.2.3	Carte des contraintes souples . . . . .	124
5.3	Proposition d'une stratégie optimale : résultats du solveur . . . . .	127
5.3.1	La stratégie optimale : compromis des différentes contraintes . . . . .	127
5.3.2	Performances . . . . .	129
5.4	Adaptation du résultat aux modifications des données . . . . .	130
5.5	Logiciel de planification : conclusions et perspectives . . . . .	133
	<b>Conclusion</b>	<b>137</b>
	<b>Liste des publications personnelles</b>	<b>141</b>
	<b>Bibliographie</b>	<b>143</b>
	<b>Annexes</b>	<b>149</b>



# Table des figures

1.1	Rendu volumique à partir d'une acquisition scanner . . . . .	8
1.2	Visualisation par reconstruction de maillages 3D . . . . .	9
1.3	Planification pré-opératoire à partir de coupes scanner 2D . . . . .	9
1.4	Simulateur d'intervention de chirurgie mini-invasive . . . . .	11
1.5	Réalité augmentée . . . . .	12
1.6	Ablation de tumeur par radiofréquence . . . . .	13
2.1	Processus de conception : modeleur déclaratif vs modeleur classique . . . . .	33
2.2	Méthodes d'optimisation . . . . .	38
2.3	Méthode de Nelder-Mead . . . . .	40
2.4	Pareto-optimalité . . . . .	44
2.5	Méthode de pondération . . . . .	47
2.6	Méthode $\epsilon$ -contrainte . . . . .	47
3.1	Représentation graphique des termes . . . . .	57
3.2	Exemples de terme . . . . .	58
3.3	Conversion contraintes strictes / contraintes souples . . . . .	60
3.4	Espace des solutions . . . . .	65
3.5	Exemple de stratégies à éliminer de l'espace des solutions . . . . .	72
3.6	Représentation des contraintes stricte . . . . .	74
3.7	Espace des solutions . . . . .	80
4.1	Evaluation d'un terme . . . . .	88
4.2	Exemple de scénario d'évaluations d'un terme . . . . .	90
4.3	Opérateur visible . . . . .	92
4.4	Détermination de la visibilité d'un triangle dans une vue . . . . .	93
4.5	Détermination de la visibilité depuis un volume . . . . .	94
4.6	Volume de nécrose correspondant à un placement d'aiguille . . . . .	97
4.7	Limite de la représentation de l'espace des solutions . . . . .	98
4.8	Subdivision des triangles pour la détermination de la zone d'insertion . . . . .	101
4.9	Subdivision du maillage de la zone d'insertion . . . . .	102
4.10	Optimisation locale sur une fonction objectif avec minima locaux . . . . .	104
4.11	Zones d'intérêt pour l'étape d'opimisation locale . . . . .	106
4.12	Convexité de la fonction associant placement d'aiguille et taille de la zone de nécrose incluant la tumeur . . . . .	110

---

5.1	Structure logiciel de planification d'ablation par radiofréquence . . . . .	117
5.2	Diagramme de classes simplifié du logiciel RFSim . . . . .	118
5.3	Exemple de graphe de scène partiel . . . . .	121
5.4	Visualisation d'un patient virtuel en 3 dimensions . . . . .	122
5.5	Visualisation simultanée du patient en 3D et des coupes scanner 2D . . .	123
5.6	Représentation de la zone d'insertion . . . . .	124
5.7	Cartes représentant le respect des contraintes souples . . . . .	126
5.8	Conséquences du changement de valeur d'une donnée par l'utilisateur . .	132
5.9	Modification de la valeur des données complémentaires . . . . .	134

# Introduction

DANS de nombreux domaines de la médecine interventionnelle, l'apport des nouvelles technologies prend une place grandissante. Les techniques d'imagerie fournissent des informations de plus en plus détaillées sur l'anatomie et la physiologie des patients. La puissance de calcul des machines actuelles et les techniques élaborées d'intelligence artificielle permettent aux médecins d'aborder les interventions en ayant une connaissance précise des meilleures possibilités de traitement et de leurs conséquences. La connaissance des phénomènes physiques impliqués dans les interventions est utilisée pour les simuler de façon réaliste et ainsi prédire les effets des traitements. Les gestes chirurgicaux peuvent être assistés et corrigés par l'utilisation de la robotique. L'utilisation de ces technologies améliore la précision et l'efficacité des traitements classiques en chirurgie. Ces technologies ont également ouvert la voie à des interventions d'un nouveau genre : les opérations de chirurgie mini-invasive. Ces interventions, d'applications variées (cancérologie, cardiologie, gynécologie, etc), ont en commun la limitation des incisions d'accès aux sites chirurgicaux et entraînent par conséquent l'impossibilité d'une visualisation directe des outils ainsi que des organes internes impliqués. Le geste du chirurgien repose complètement sur les informations visuelles disponibles via une caméra insérée à l'intérieur du site chirurgical ou des acquisitions d'imagerie par voie externe.

Le confort du patient implique alors des difficultés supplémentaires pour le chirurgien. La précision du geste dans ce type d'intervention est d'autant plus importante que la marge de manœuvre du praticien est limitée, celui-ci opérant par l'intermédiaire d'outils chirurgicaux insérés à l'intérieur du patient. Le principe de ces techniques repose dès l'origine sur l'utilisation d'outils d'imagerie médicale sophistiqués. Elles s'accompagnent de plus en plus de l'utilisation de robots qui permettent de compenser la perte de dextérité liée à la limitation des accès au site chirurgical. Les données d'imagerie numérique indispensables à l'intervention peuvent également être davantage exploitées par des procédés informatiques afin d'alléger la tâche du praticien. La préparation d'une intervention de chirurgie mini-invasive repose sur une étude minutieuse des images pré-opératoires qui renferment la plupart des informations visuelles disponibles pour l'intervention. Cette étude aboutit à la détermination d'une stratégie complète pour l'opération impliquant le choix des outils chirurgicaux adéquats, des points d'accès optimaux, du dosage du traitement le cas échéant, etc. La décision du praticien repose en grande partie sur son expérience propre ainsi que sur les conclusions des études médicales récentes du domaine. La perception des différentes stratégies possibles pour chaque intervention ainsi que l'appréciation de leurs valeurs relatives nécessite un apprentissage spécifique et une expérience solide de l'intervention considérée. En supposant que les contraintes régissant la planification pré-opératoire

d'une intervention mini-invasive particulière soient connues, nous pensons qu'il est possible d'assister cette tâche par ordinateur grâce à un programme capable de les résoudre en fonction de l'anatomie spécifique du patient considéré.

L'utilisation d'un tel outil logiciel pourrait s'avérer précieux pour le choix d'une stratégie opératoire en particulier pour les médecins novices en chirurgie mini-invasive. Cependant, la majorité des interventions mini-invasives sont récentes et font l'objet de progrès technologiques en permanence : perfectionnement des outils chirurgicaux, enrichissement des techniques, émergence de nouvelles procédures plus efficaces. Si l'on ne tient pas compte du caractère dynamique de ces nouvelles techniques chirurgicales lors du développement de l'outil de planification, celui-ci deviendra rapidement dépassé et donc inutilisable. Il est possible de considérer les règles de planification, non pas comme ancrées dans le logiciel de planification, mais comme des données indépendantes, modifiables, tout comme les données anatomiques du patient. Si l'on considère les règles de planification définies pour un type d'intervention particulier, le logiciel a alors le rôle d'un solveur de contraintes capable de les appliquer sur une anatomie de patient quelconque.

Dans le cadre de cette thèse, nous nous intéressons à la planification pré-opératoire d'un ensemble de techniques de chirurgie mini-invasive en plein essor dans le domaine de l'oncologie : les interventions de thermoablation de tumeurs. Elles consistent en une destruction des tumeurs in situ par l'effet nocif de températures extrêmes généralement délivrées par l'intermédiaire de sondes spécifiques. L'objectif de ces travaux est de proposer un outil d'assistance efficace, intuitif et facilement adaptable pour ce type d'intervention. L'approche que nous avons adoptée est de considérer l'outil de planification comme un outil de collaboration entre l'expert qui définit les règles de planification, et la machine qui les résout. Cela implique qu'il est nécessaire de développer des outils de communication homme/machine en complément des méthodes de résolutions du problème de planification. Nos travaux sont donc articulés selon trois axes :

- le développement d'outils de formalisation des règles de planification manipulables par l'expert et utilisables par la machine,
- la construction d'un solveur de contraintes efficace,
- la réalisation d'une interface intuitive permettant un accès facile aux informations de résolution et une réactivité de la planification.

Dans le premier chapitre, nous introduisons le contexte médical dans lequel se placent nos travaux. Après une présentation succincte des différents axes de recherches liés à l'assistance informatisée aux opérations mini-invasives, nous exposons le principe des opérations d'ablations de tumeurs qui appartiennent au champ d'application de nos travaux. Nous détaillons en particulier la technique d'ablation par radiofréquence sur laquelle nous avons concentré nos investigations. Puis nous dressons un état de l'art des approches existantes dans le domaine de l'informatique concernant la planification préopératoire des interventions mini-invasives.

Le second chapitre traite de deux domaines vastes et largement explorés en informatique : la résolution de contraintes et l'optimisation. Concernant la résolution de contraintes, on s'intéresse en particulier à la résolution de contraintes géométriques puisque, dans notre

cas, il s'agit de planifier un geste chirurgical adapté à la géométrie associée à l'anatomie propre à chaque patient. Pour cela, nous nous sommes inspirés des études existantes dans le domaine de la CAO, et en particulier de l'approche de *modélisation déclarative* qui s'efforce de proposer des outils de modélisation de haut niveau pour rendre la tâche de conception plus naturelle et plus intuitive. Dans ce chapitre nous examinons également la problématique de l'optimisation difficile. Après un tour d'horizon de quelques approches de résolution classiques pour les problèmes considérant une fonction objectif unique, nous présentons les particularités des problèmes multi-critères, pour lesquels le concept même d'optimum doit être redéfini.

Les chapitre 3, 4 et 5 exposent les contributions apportées par cette thèse concernant la problématique d'assistance à la planification d'opérations de thermoablation.

Afin de formaliser les règles de planification, nous avons développé un langage de définition de contraintes que nous détaillons dans le chapitre 3. Ce langage permet d'ajouter des données complémentaires aux données anatomiques, utiles à la planification, ainsi que de définir des contraintes géométriques à respecter ou à optimiser. Nous proposons également en exemple un script illustrant l'utilisation de ce langage, dans le cas de la planification des ablations par radiofréquence pour les tumeurs hépatiques.

Le chapitre 4 détaille les techniques de résolution développées spécifiquement pour la résolution de contraintes quelconques, définies dans le cadre décrit dans le chapitre 3. Le processus de résolution des contraintes comprend plusieurs mécanismes : l'évaluation des opérations géométriques associées aux contraintes ainsi leur traitement selon la nature des contraintes correspondantes. Alors que les contraintes à respecter mènent à la construction d'un espace des solutions, les contraintes à optimiser permettent d'ordonner les possibilités et de sélectionner une stratégie opératoire parmi ces dernières.

Le dernier chapitre est consacré à la présentation de l'outil de planification d'un point de vue logiciel. En particulier, nous détaillons les outils d'interaction disponibles à travers l'interface graphique, permettant à l'utilisateur d'accéder facilement aux informations de planification, et de l'adapter si nécessaire.



# Chapitre 1

## Contexte médical

### Sommaire

---

<b>1.1</b>	<b>Apport de l'informatique aux opérations de chirurgie mini-invasives . . . . .</b>	<b>6</b>
1.1.1	Les interventions chirurgicales minimalement invasives . . . . .	6
1.1.2	Techniques d'imagerie médicale . . . . .	6
1.1.3	Visualisation 3D . . . . .	7
1.1.4	Aide à la décision . . . . .	9
1.1.5	Simulation . . . . .	10
1.1.6	Assistance per-opératoire . . . . .	10
<b>1.2</b>	<b>Thérapies minimalement invasives pour l'ablation de tumeurs</b>	<b>12</b>
1.2.1	La thermo-ablation par radiofréquence . . . . .	13
1.2.1.1	Principe . . . . .	13
1.2.1.2	Indications . . . . .	14
1.2.1.3	Zone de coagulation conséquent au traitement . . . . .	15
1.2.1.4	Etape de planification : choix d'une stratégie d'intervention . . . . .	16
1.2.2	Autres types d'ablation minimalement invasives . . . . .	18
1.2.2.1	Cryoablation . . . . .	18
1.2.2.2	Alcoolisation . . . . .	18
1.2.2.3	Autres opérations de thermo-ablation . . . . .	19
<b>1.3</b>	<b>Planification pré-opératoire assistée par ordinateur . . . . .</b>	<b>20</b>
1.3.1	Simulation du traitement . . . . .	20
1.3.2	Aide à la décision . . . . .	22
1.3.2.1	Cryochirurgie de la prostate . . . . .	22
1.3.2.2	Thermo-ablations de tumeurs hépatiques . . . . .	23
1.3.2.3	Interventions minimalement invasives robotisées . . . . .	23
1.3.2.4	Problématiques communes aux différentes interventions . . . . .	24
1.3.2.5	Positionnement de ces travaux . . . . .	24

---

## 1.1 Apport de l'informatique aux opérations de chirurgie mini-invasives

### 1.1.1 Les interventions chirurgicales minimalement invasives

LORS d'une intervention chirurgicale, le chirurgien doit pouvoir accéder librement aux tissus et organes situés parfois profondément dans le corps humain. La méthode classique consiste en une large incision et un dégagement du site permettant un accès aisé aux organes concernés. Malheureusement, les opérations de chirurgie ouverte sont très traumatisantes pour le patient ; elles entraînent un temps de récupération important et un risque de complications relativement élevé. Ces opérations peuvent donc être contre-indiquées pour certains patients qui ne sont pas physiquement aptes à supporter une telle procédure. Depuis une vingtaine d'années, une nouvelle chirurgie apparaît et s'impose progressivement comme une nécessité d'évolution : la chirurgie mini-invasive. Ces opérations limitent les traumatismes inutiles soit en effectuant des incisions d'accès de l'ordre du centimètre, soit en passant par les voies naturelles (opération Anubis, IRCAD, Avril 2007). La chirurgie mini-invasive présente donc de multiples avantages : possibilité de traiter des patients qui ne supporteraient pas la chirurgie classique, minimisation du traumatisme liée à l'intervention, séjour hospitalier court et réduction du coût du traitement. Cependant, ces techniques impliquent des difficultés supplémentaires pour le praticien :

- la visualisation des organes internes n'est pas directe lors des interventions. Elle n'est possible que par les procédés classiques d'imagerie médicale, ou grâce à un endoscope qui fournit des images non stéréoscopiques, qui peuvent présenter des déformations géométriques et sont limitées en résolution, en contraste et en couleur.
- les sensations tactiles sont limitées parce que le praticien interagit avec les organes internes au moyen d'instruments chirurgicaux qui se trouvent à l'extrémité de longs tubes fins
- le point d'insertion de l'outil contraint les déplacements possibles à l'intérieur du patient, seules les rotations centrées au point d'insertion ou les translations suivant l'axe de l'outil sont envisageables.
- la coordination main-œil est difficile car la contrainte du point pivot de l'outil intervertit les directions de mouvement à l'intérieur et à l'extérieur du corps.

Ces inconvénients augmentent le temps d'exécution des procédures et les risques d'erreur rendant les techniques de chirurgie mini-invasive particulièrement difficiles à mettre en œuvre, nécessitant ainsi un apprentissage spécifique.

### 1.1.2 Techniques d'imagerie médicale

L'essor des techniques de chirurgie mini-invasive repose principalement sur les progrès des procédés d'imagerie qui fournissent des acquisitions de l'anatomie interne du patient de plus en plus précises et informatives. Il existe différents procédés d'imagerie médicale, nous ne citons ici que les principales techniques servant au diagnostic, à la planification et



au guidage des opérations de chirurgie mini-invasive :

- l'imagerie par ultrasons (échographie) qui permet de visualiser en temps réel une coupe anatomique du patient grâce à une sonde déplacée sur la surface de la peau. Cette méthode est totalement inoffensive pour le patient et peu coûteuse, cependant les images obtenues ne sont pas toujours très nettes. De plus le positionnement manuel de la sonde varie d'un examen à l'autre et n'est pas connu *a priori*, ce qui rend complexe toute réinterprétation de l'examen ou éventuel recalage avec d'autres modalités d'imagerie médicale.
- l'imagerie scanner par rayon X (tomodensitométrie) qui permet de visualiser une série de coupes parallèles rapprochées représentant une partie de l'anatomie du patient à un instant donné. Cette technique permet de faire ressortir certains tissus, notamment le système vasculaire, grâce à l'injection de produit de contraste avant l'acquisition. Cependant, le scanner est coûteux et encombrant ce qui rend l'accessibilité à ce genre d'imagerie relativement limitée. De plus l'acquisition implique l'irradiation du patient qui, si l'exposition est répétée, peut être nocive pour l'organisme.
- l'imagerie par résonance magnétique nucléaire (IRM) qui permet d'obtenir des images comparables aux images scanner. Cette technique est particulièrement performante pour l'observation des tissus mous (en particulier le cerveau par exemple). Contrairement à l'imagerie scanner, l'examen IRM n'irradie pas. Cependant, elle nécessite également du matériel coûteux et encombrant qui n'est pas forcément accessible.

Ces images sont utilisées par le radiologue interventionnel ou le chirurgien lors des différentes étapes d'une opération de chirurgie mini-invasive (planification pré-opératoire, opération, suivi). L'utilisation de ces images pourrait être valorisée par l'intégration systématique de l'assistance par ordinateur dans le processus opératoire. Dans les prochains paragraphes, nous allons détailler comment l'exploitation des images médicales numériques par des techniques de traitement d'image, d'informatique graphique, d'intelligence artificielle et de simulations physiques informatisées peut faire bénéficier les interventions mini-invasives d'une assistance informatique précieuse, aussi bien au niveau de la planification pré-opératoire et de l'aide à la décision qu'au moment du geste opératoire en lui même.

### 1.1.3 Visualisation 3D

Les systèmes d'imagerie médicale classiques permettent de visualiser l'anatomie interne des patients par coupes bi-dimensionnelles. La lecture de telles images nécessite un certain apprentissage avant de parvenir à se repérer dans les images et recalculer mentalement les structures 2D visualisées par rapport au patient 3D. Des méthodes ont donc été développées pour permettre de visualiser plus intuitivement l'anatomie 3D du patient à partir des données issues des acquisitions d'imagerie médicale. On peut les classer en 2 catégories :

- la visualisation par rendu volumique
- la visualisation par maillage

La visualisation par rendu volumique consiste simplement à projeter sur chaque pixel du plan de vue les informations correspondantes contenues dans le volume formé par l'ensemble des images d'une acquisition. Chaque niveau de gris d'une image correspond à une certaine

couleur et une certaine transparence, c'est ce qu'on appelle la fonction de transfert. Elle permet par exemple de voir en transparence les organes et de les différencier par des couleurs (voir figure 1.1). Cette méthode ne nécessite aucune connaissance particulière sur l'anatomie, elle permet de visualiser directement les données et de mettre en évidence les organes d'intérêt en agissant sur la fonction de transfert.

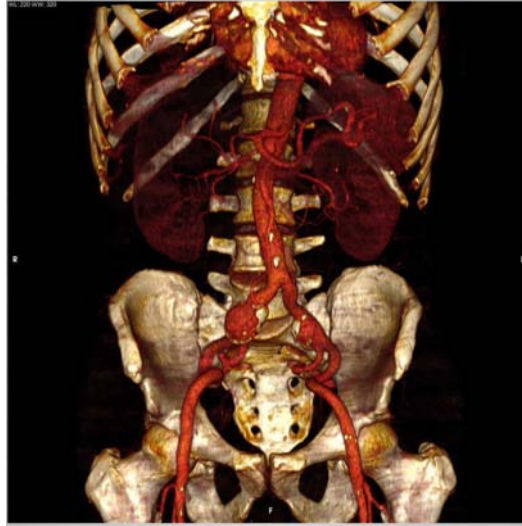


FIGURE 1.1: Rendu volumique des données anatomiques issues d'une acquisition scanner

La visualisation par maillage n'utilise pas directement les données issues de l'acquisition. Elle passe par deux étapes de pré-traitement aboutissant à la création de maillages 3D représentant chaque organe du patient qui sont alors visualisés comme des données 3D classiques [BP07]. La première étape est la segmentation, elle consiste à étiqueter chaque voxel de l'image comme faisant partie ou non d'un organe à visualiser. De nombreux travaux ont été effectués dans le domaine du traitement d'image pour faciliter cette étape de segmentation, voire même de l'automatiser [SDM01]. Cependant le traitement complètement informatisé de cette étape est encore délicat dans la mesure où même le résultat de l'étiquetage manuel des images scanner peut varier d'un radiologue à l'autre. La deuxième étape est la construction d'un maillage représentant l'enveloppe surfacique d'un ensemble de voxels identifié par la première étape comme une structure anatomique distincte. Il existe des méthodes de construction automatique de maillages triangulaires à partir des données voxels volumiques, cependant il se pose le problème de la qualité des surfaces reconstruites avec un nombre de polygones raisonnable permettant de limiter les calculs par la suite, mais également la question de la cohérence entre les éléments 3D (inclusion ou voisinage de deux structures anatomiques par exemple) [BBC<sup>+</sup>09]. Grâce à la reconstruction 3D des différentes structures anatomiques du patient, la visualisation indépendante de chacun des organes est possible, et la mise en évidence ou en parenthèse de certaines structures est possible grâce à la propriété de transparence des maillages représentant les organes (voir figure 1.2). Cette méthode de visualisation nécessite par contre un prétraitement impor-

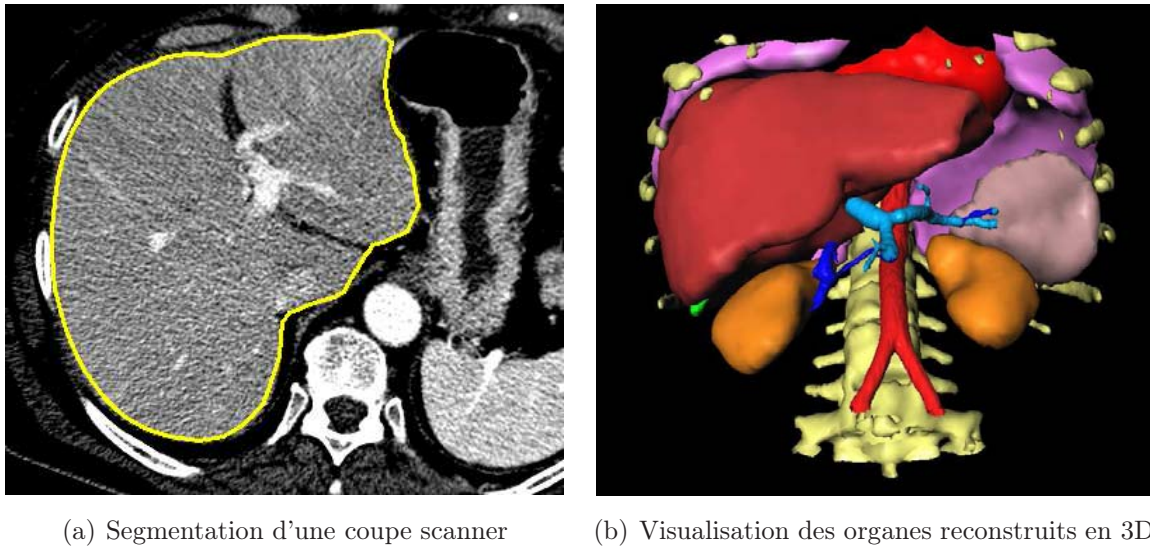


FIGURE 1.2: Visualisation par reconstruction de maillages 3D

tant impliquant l'apport de connaissances anatomiques pour l'étape de segmentation, mais celle-ci est indispensable à l'étape qui nous intéresse plus particulièrement : l'aide à la décision.

La visualisation tri-dimensionnelle des structures anatomiques et pathologiques apporte une vue d'ensemble plus intuitive de chaque cas et peut être couplée avec la visualisation classique par coupe pour faciliter l'étape de planification pré-opératoire.

#### 1.1.4 Aide à la décision



FIGURE 1.3: Planification pré-opératoire à partir de coupes scanner 2D

L'aide à la décision est le sujet principal de ce mémoire. Classiquement le médecin se fonde sur les coupes bidimensionnelles issues de l'acquisition pour choisir ses outils, leurs positionnements relatifs et les modalités du traitement (temps d'exposition à la chaleur

par exemple dans le cas d'une thermo-ablation). Cette visualisation par coupes 2D peut s'avérer insuffisante pour placer précisément les outils de chirurgie mini-invasive dans le site chirurgical et donc limiter l'efficacité du traitement [AKV<sup>+</sup>02]. Le modèle 3D de l'anatomie du patient et de sa pathologie permet de faciliter la compréhension du cas à traiter grâce à une visualisation intuitive, mais il peut également être exploité à d'autres fins. Notamment il peut être interprété de façon à calculer un certain nombre d'informations qui sont susceptibles d'intéresser le médecin et entrer en ligne de compte dans le choix d'une stratégie opératoire. On peut aller plus loin dans l'aide à la décision en proposant automatiquement des solutions adaptées à chaque cas, que le médecin pourra choisir de valider, d'adapter ou de rejeter. Cela nécessite de connaître et de rassembler, pour chaque type d'intervention, les critères déterminants d'un choix de stratégie opératoire. Nous détaillerons dans la section 1.3.2 les travaux existants dans le domaine de l'assistance informatique à la planification pré-opératoire.

### 1.1.5 Simulation

Les reconstructions de modèle 3D du patient à partir des images pré-opératoires peuvent également être exploitées afin de simuler virtuellement l'intervention. Cette approche présente plusieurs intérêts. Le premier intérêt est d'ordre pédagogique. L'utilisation de simulateurs permet aux apprentis chirurgiens d'améliorer leur dextérité et leur maîtrise des outils de chirurgie mini-invasive dans des conditions virtuelles les plus proches possibles des conditions réelles [BSHW07]. Les simulateurs informatiques leur permettent donc d'acquérir une certaine expérience sur de multiples interventions virtuelles avant de l'appliquer en conditions réelles et d'être plus efficace dès leurs premières interventions sur des patients réels. Le deuxième intérêt réside dans la possibilité d'effectuer une répétition virtuelle de l'opération avant d'appliquer la stratégie opératoire sélectionnée en salle d'opération. Les avantages sont alors multiples :

- détection des difficultés spécifiques à l'intervention,
- simulation des effets du traitement correspondant à une stratégie donnée et estimation de son succès,
- adaptation possible de la stratégie opératoire finale au regard de cette expérience en cas d'échec ou de succès mitigé.

Dans les deux cas, la visualisation réaliste du patient virtuel ainsi que la simulation des déformations des organes provoquée par la manipulation des outils virtuels [CDA99] peut-être couplée avec l'utilisation de périphériques à retour d'effort permettant de reproduire la sensation tactile ressentie lors de l'opération mais également avec une interface de travail semblable à la réalité avec patient artificiel et manipulation de vrais outils chirurgicaux [BGT<sup>+</sup>96] (voir figure 1.4) .

### 1.1.6 Assistance per-opératoire

Le développement de la chirurgie mini-invasive implique de nouvelles difficultés qui peuvent être surmontées en mettant en place de nouveaux protocoles d'intervention per-



FIGURE 1.4: **Simulateur d'intervention de chirurgie mini-invasive** : Les outils chirurgicaux sont inséré à travers un périphérique haptique permettant de reproduire les sensations tactiles.

mettant d'assister le chirurgien. L'exploitation des images médicales par des procédés informatiques peut intervenir jusque dans la salle d'opération et faire bénéficier le praticien d'outils précieux pour mettre en œuvre de manière efficace sa stratégie d'intervention. En effet, s'il est relativement simple pour le praticien de placer dans une scène 3D des outils de chirurgie mini-invasive lors de la phase de planification pré-opératoire, la reproduction de ce geste planifié peut s'avérer délicat. L'utilisation des informations issues de la phase de planification, visibles sur un écran, se heurte à la difficulté de les mettre en correspondance avec la vision directe du patient, dont les points de repère sont limités à quelques repères anatomiques externes dans le cas de la chirurgie mini-invasive (peau, côtes dans le cas d'interventions abdominales). Le praticien doit donc réaliser un repositionnement mental dont la qualité va dépendre de sa capacité à se servir de repères anatomiques du patient. Cette tâche abstraite limitant la précision du geste peut être facilitée par des techniques de réalité augmentée permettant de visualiser simultanément la réalité et les images issues de la planification mises en correspondance [Nic04] (voir figure 1.5). Dans le cas d'interventions abdominales, la reproduction du geste planifié est d'autant plus délicate dans la réalité que les organes ciblés bougent constamment à cause de la respiration du patient alors que les images sont statiques. Les techniques de réalité augmentée peuvent alors guider le praticien vers une cible mobile en compensant ce mouvement respiratoire [MHPA<sup>+</sup>07]. La problématique de la reproduction d'un geste planifié de façon virtuelle dans la salle d'opération se pose également dans le domaine de la chirurgie mini-invasive robotisée [CMAMB04]. L'assistance robotique lors d'interventions mini-invasives permet d'augmenter la dextérité du praticien en surpassant les limites imposées par la physiologie humaine (nombre de bras, degrés de libertés des outils, possibilité de compenser les mouvements indésirables) et rencontre donc un succès grandissant.



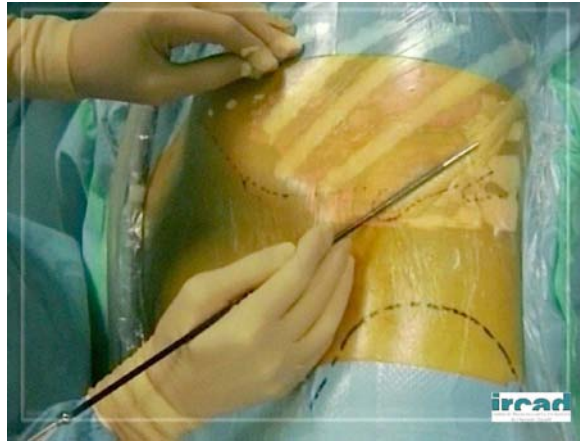


FIGURE 1.5: Réalité augmentée : superposition de la reconstruction 3D avec les images issues d'une caméra dans le bloc opératoire

## 1.2 Thérapies minimalement invasives pour l'ablation de tumeurs

Il existe une grande variété d'opérations mini-invasives qui s'appliquent à différents domaines de la médecine. Dans le cadre de ce mémoire, nous nous intéressons plus particulièrement à l'ablation de tumeurs cancéreuses. Le traitement par voie chirurgicale classique consiste à ouvrir le patient et retirer les portions d'organe atteintes. Cette approche reste à l'heure actuelle la plus connue et la plus pratiquée lorsque le cas le permet. Les opérations mini-invasives d'ablation locales offrent une alternative intéressante. En plus des avantages accompagnant leur caractère mini-invasif, ces nouvelles techniques permettent de proposer un traitement potentiellement curatif à une plus large proportion de patients. En effet la chirurgie classique comporte beaucoup de contre-indications qui peuvent concerner l'état général du patient, la localisation ou la dispersion des tumeurs à traiter par exemple. La thérapie d'ablation locale peut alors intervenir soit pour traiter toutes les tumeurs concernées, soit en complément d'une opération chirurgicale. De plus, la durée d'hospitalisation conséquente à ce type d'opération est réduite, tout comme les coûts induits. Les techniques mini invasives d'ablation de tumeur développées depuis ces vingt dernières années sont variées et connaissent régulièrement des avancées techniques permettant de les considérer actuellement comme un traitement à visée curative prometteur [GCC<sup>+</sup>03, Kud04]. Ces opérations consistent à détruire les tumeurs *in situ* en dénaturant les cellules tumorales par l'effet de la température ou de produits toxiques. Dans le cas des opérations de thermo-ablation, l'objectif est d'atteindre une température létale dans les tissus cancéreux ciblés tout en préservant au maximum les tissus sains. La majorité de ces opérations ont été expérimentées principalement pour des tumeurs hépatiques, mais concernent de plus en plus d'autres localisations de tumeurs. Puisque nous nous intéressons à la planification automatique de ce type d'opérations, il est nécessaire de rassembler l'expertise médicale dans ce domaine. Dans cette partie, nous allons donc détailler en particulier l'ablation par

radiofréquence appliquée à des tumeurs hépatiques principalement, cette technique étant la plus prometteuse et la plus documentée à l'heure actuelle. Puis nous nous intéresserons aux autres techniques voisines pour lesquelles ce travail de thèse n'a pas été spécifiquement développé mais pourrait être facilement adapté.

## 1.2.1 La thermo-ablation par radiofréquence

### 1.2.1.1 Principe

L'ablation par radiofréquence (Radio Frequency Ablation RFA) consiste à placer la pointe d'une aiguille à l'intérieur de la tumeur de façon à dispenser la chaleur provoquant la nécrose des tissus cancéreux par brûlure. L'opération peut être effectuée :

- lors d'une opération chirurgicale ouverte : cette voie est utilisée lorsque le traitement est couplé avec une résection hépatique par exemple. L'accès à la tumeur est facilité mais l'opération perd son caractère mini-invasif.
- par voie laparoscopique : des petites incisions sont effectuées dans la peau du patient de façon à pouvoir insérer des trocars (guides) dans lesquels on fera passer les outils (dont une caméra) dans l'abdomen.
- par voie percutanée : l'aiguille est simplement insérée à travers la peau du patient, son placement est guidé par l'une des méthodes d'imagerie décrites précédemment.

Dans le cadre de ce mémoire nous étudierons l'ablation par voie percutanée qui est la moins invasive, mais également celle pour laquelle l'assistance informatique est la plus appropriée, que ce soit de façon pré ou per-opératoire. En effet le manque de visibilité et de possibilité d'ajustement du placement des outils pendant l'opération conduit à consacrer une attention particulière à la phase de planification de cette thérapie.

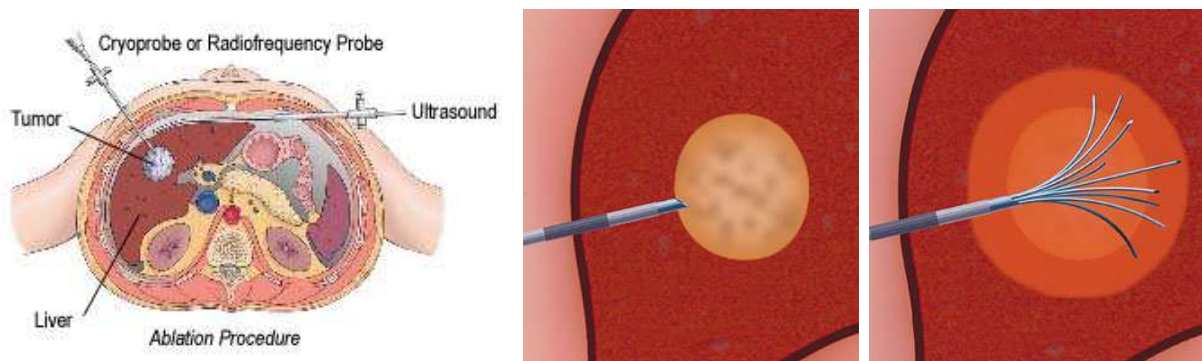


FIGURE 1.6: **Ablation de tumeur hépatique par radio-fréquence** : L'aiguille est insérée jusqu'au foie à travers la peau. Dans le cas représenté il s'agit d'une aiguille à ombrelles, de multiples électrodes sont alors déployées. A la fin du traitement un volume comprenant la tumeur et une marge de tissus sains est brûlé.

Lors de ce type d'opération, la tumeur est détruite grâce à l'effet thermique du passage de courant électrique dans le corps du patient. Un circuit électrique fermé est créé, consti-

tué d'un générateur de courant alternatif de radiofréquence<sup>1</sup>, une aiguille (ou électrode) adaptée, insérée dans la tumeur, le patient et des plaques dispersives placées généralement sur ses cuisses. Le passage du courant alternatif provoque une agitation ionique dans les tissus, ce qui a pour effet d'augmenter la température cellulaire. Les plaques dispersives, dont la surface de contact avec le corps est délibérément importante en comparaison de celle de la pointe active de l'aiguille, ont pour rôle de diminuer la densité du courant à distance de celle-ci et ainsi de provoquer une augmentation ciblée de la température à proximité de la pointe de l'aiguille. La nature des dommages thermiques subis par les tissus dépend de la température atteinte mais aussi du temps d'exposition à la chaleur. Une température de 50°C provoque des dommages irréversibles en 4 à 6 minutes, à partir de 60°C la coagulation est pratiquement immédiate. Au delà de 100°C les tissus sont carbonisés et des bulles de vapeur se forment, ce qui a pour effet de diminuer la conductivité thermique dans les tissus et peut conduire à une ablation incomplète. L'objectif est donc d'atteindre une température entre 50 et 100°C pendant 4 à 6 minutes dans tout le volume à détruire. Plusieurs articles présentent le principe de l'opération de radiofréquence de manière plus détaillée [MD01, Gol01, RDC<sup>+</sup>04].

### 1.2.1.2 Indications

La RFA est utilisée depuis une vingtaine d'années pour détruire des tumeurs hépatiques. Plus récemment cette technique s'est développée pour le traitement de tumeurs rénales et pulmonaires, et de façon plus anecdotique pour d'autres localisations tumorales (os, seins). Nous nous sommes concentrés sur la littérature médicale traitant des ablations hépatiques bien que la plupart de ces informations ne soient pas spécifiques à cet organe. Les indications de la technique peuvent différer d'un centre de traitement à l'autre, cependant un consensus semble établi quant aux cas pour lesquels la RFA doit être la technique à privilégier [Cur03, WRC<sup>+</sup>00] :

- Les patients candidats à la RFA sont ceux pour lesquels la chirurgie n'est pas envisageable. Comme nous l'avons déjà évoqué plus tôt les contre-indications à la chirurgie sont multiples, ce qui se traduit par une proportion de patients opérables limitée (de l'ordre de 20% dans les cas de métastases hépatiques). La RFA peut alors être envisagée pour la grande proportion des patients restants.
- La technique est pour l'instant limitée avant tout par la taille des tumeurs qu'elle permet de traiter et par l'incertitude sur la destruction totale ou la survie résiduelle de tissus tumoraux qui auraient dû être détruits. Les meilleures chances d'ablation totale existent pour les tumeurs de moins de 4 cm sans contact avec les grands vaisseaux. Elle est toutefois utilisée pour des tumeurs de plus grande ampleur, avec un taux de réussite diminuant fortement avec l'augmentation du diamètre tumoral.

Les évolutions techniques des générateurs et aiguilles de radiofréquence permettront d'intervenir sur des volumes tumoraux de plus en plus grands, si bien que le facteur limitant de la taille des tumeurs, perdra certainement de son importance dans le futur. De plus, les

---

1. L'utilisation de fréquences élevées élimine le risque de tétanie, potentiellement létale



nouveaux procédés d'imagerie permettront peut-être de se prononcer sur l'intégralité de l'ablation tumorale pendant l'intervention ou immédiatement après, ce qui se répercuterait de façon positive sur le taux de succès de cette technique et donc sur son utilisation plus massive.

### 1.2.1.3 Zone de coagulation consécutive au traitement

Les cas de réapparition de tumeur sont fortement liés à la taille et la forme de la zone de nécrose induite par le traitement (que l'on appellera également *lésion*) puisqu'il est difficile d'une part de prédire à l'avance quels vont être les tissus effectivement détruits; d'autre part, la destruction de grosses tumeurs nécessite souvent plusieurs ablations ce qui augmente considérablement les risques d'imprécision dans les placements des électrodes. La détermination de la géométrie de la lésion ainsi que l'augmentation de sa taille ont été l'objet d'un intérêt particulier pour l'amélioration des résultats de cette intervention.

La lésion de radiofréquence peut avoir différentes formes et tailles selon le matériel utilisé (générateur et type d'aiguille) et le temps d'application du traitement. Diverses expérimentations ont été effectuées, principalement *in vivo* dans des foies de porcs afin de déterminer les tailles et formes de lésion correspondant à différents modèles d'électrodes. Mulier *et al* recense un certain nombre de ces études dans [MNM<sup>+</sup>03]. La forme des lésions est généralement proche d'un ellipsoïde (voire d'une sphère dans certains cas) dont la taille peut varier entre 1 et 5,5 cm pour les modèles d'aiguilles présentés dans l'article. Cependant ces études manquent de précisions sur la description de la géométrie de la lésion. En effet la plupart d'entre elles précisent uniquement son diamètre, considérant que la lésion est une sphère, alors que des informations comme la position relative de la lésion par rapport à l'aiguille ou le degré d'ellipticité ont leur importance [MNJ<sup>+</sup>05]. Notons que la forme de la lésion peut également être influencée par la nature des tissus dans lesquels on place l'aiguille. La principale cause de perte de chaleur à proximité de l'aiguille est la perfusion des tissus. Ainsi, si l'on tente une ablation à proximité d'un gros vaisseau sanguin, les tissus proches sont protégés par le refroidissement dû au flux sanguin [SSP<sup>+</sup>99]. La lésion résultante est donc déformée par rapport au modèle classique ellipsoïdal. Des méthodes de simulation physique par informatique ont été développées de façon à prédire les effets du traitement de façon plus précise (cf. section 1.3.1).

La principale limite de l'utilisation de la radiofréquence est la taille des tumeurs traitables par une ablation unique. Cette limite a été repoussée grâce à différentes techniques [Gol01]. Une première approche s'appuie sur le développement d'aiguilles de radiofréquence de plus en plus sophistiquées (aiguilles à électrodes multiples, bipolaires, à refroidissement interne) [CSD<sup>+</sup>07]. Une autre approche consiste à agir sur les propriétés des tissus, soit en augmentant leur conductivité thermique et électrique (injection de solution saline), soit en diminuant la perte de chaleur due à la perfusion des tissus (occlusion mécanique de gros vaisseaux ou diminution pharmacologique du flux sanguin). Lorsque la tumeur est trop volumineuse pour être complètement englobée dans la zone de nécrose, il est possible de replacer une ou plusieurs fois l'aiguille à différents endroits de la tumeur de façon à détruire tous les tissus cancéreux. Des modèles d'ablations multiples ont été proposés

[DFA<sup>+</sup>01][KSSH04], cependant même avec un nombre important d’ablations superposées, la taille des tumeurs traitables augmente peu, alors que les risques d’erreur de placement se multiplient. La forme compliquée et difficilement prévisible présentée par la superposition des brûlures conduit donc plus fréquemment à une destruction incomplète des cellules cancéreuses ciblées et accroît la proportion de récurrence. L’augmentation de la taille de la lésion en une ablation unique permet alors d’envisager beaucoup plus de tumeurs comme potentiellement traitables par radiofréquence.

#### 1.2.1.4 Etape de planification : choix d’une stratégie d’intervention

L’ablation par radiofréquence tout comme la plupart des opérations de chirurgie mini-invasive nécessite un temps d’apprentissage relativement long avant d’être maîtrisée par le praticien [PNL<sup>+</sup>04]. Les praticiens expérimentés ont reporté les recommandations issues de leur apprentissage dans différentes études. Nous allons tout d’abord décrire le déroulement classique d’une opération de radiofréquence hépatique par voie percutanée tel que nous avons pu l’observer à l’hôpital civil de Strasbourg. Puis nous détaillerons les informations présentes dans la littérature médicale concernant les différentes stratégies recommandées pour favoriser la réussite d’une intervention.

A l’hôpital civil de Strasbourg, l’ablation par radiofréquence est guidée généralement par imagerie scanner avec quelquefois le renfort de l’échographie. Avant l’intervention, le patient est placé dans le scanner pour une première acquisition de la totalité de l’abdomen afin de vérifier l’évolution des tumeurs et de définir une stratégie spécifique. Le praticien choisit son matériel en fonction de la localisation et la taille de la tumeur puis définit une trajectoire en se basant sur les coupes 2D. Parfois il arrive qu’une deuxième acquisition complète soit demandée, soit parce que les tumeurs et les vaisseaux sanguins à proximité ne sont pas suffisamment contrastés sur l’image originale (une injection de produit de contraste est alors effectuée avant la nouvelle acquisition), soit parce que la trajectoire que le médecin envisage de prendre nécessite de changer la position du patient ou d’incliner le scanner. Une fois la trajectoire choisie sur les images pré-opératoires, le point d’entrée sélectionné est repéré sur une coupe souvent par rapport à une côte et marqué sur le patient. Le radiologue insère alors son aiguille progressivement en se guidant grâce à des petites acquisitions ciblées prises à étapes régulières sur la trajectoire, ce qui lui permet de la corriger si nécessaire et au final de vérifier que la tumeur est atteinte. Une fois l’aiguille correctement positionnée, le circuit électrique est constitué et le traitement démarre. Le générateur de radiofréquence est activé aussi longtemps qu’il est estimé nécessaire selon la taille de la tumeur. Dix à vingt minutes par ablation peuvent être requises pour s’assurer que tout le volume à détruire a atteint une température critique pendant un temps suffisamment long. Alors que l’étape de traitement en elle-même est relativement courte, la planification et la mise en place des outils peut durer assez longtemps (souvent plus d’une heure), surtout dans les cas où de multiples ablations ont été prévues.

L’objectif principal de la thérapie est de détruire la totalité des tissus cancéreux repérés sur les images pré-opératoires. Il arrive que quelques cellules cancéreuses échappent au traitement, cependant il n’est pas possible de déterminer directement après l’opération

quels sont les tissus nécrosés suite à l'opération et s'il reste des tissus tumoraux non traités. Lors du suivi post-opératoire effectué les mois suivant l'opération, il est possible dans certains cas de constater l'apparition d'une tumeur qui s'est développée à partir des cellules cancéreuses non traitées à l'emplacement initial de la tumeur. Pour éviter au maximum ce risque de récurrence des tumeurs, les spécialistes recommandent de cibler la tumeur augmentée d'une marge de tissus sains de 5 à 10 mm. D'une part, cette marge permet de compenser les petites imprécisions dues à la reproduction du geste planifié dans la réalité : le placement précis de l'aiguille dans la cible n'est pas évident et la taille de la lésion réelle peut différer de la lésion supposée. D'autre part, la frontière entre tumeur et tissus sains n'étant pas franche, les tissus entourant la tumeur non identifiés comme cancéreux à l'imagerie peuvent tout de même contenir des cellules cancéreuses.

En dehors d'une ablation incomplète de la tumeur qui signifie l'échec de l'opération lorsque celle-ci a une visée curative<sup>2</sup>, un certain nombre de complications plus ou moins fréquentes peuvent survenir à la suite d'une ablation par radiofréquence [MMN<sup>+</sup>02]. Les complications conséquentes au traitement peuvent être dues soit à l'insertion de l'électrode, soit à la thérapie thermique en elle-même. Dans la première catégorie on peut citer les hémorragies, les infections, mais aussi à plus long terme l'apparition d'une tumeur sur le trajet de l'aiguille à cause du déplacement de cellules cancéreuses ayant adhéré à l'électrode lors de son retrait. La seconde catégorie regroupe les dommages thermiques qui peuvent toucher les structures anatomiques à proximité de la zone ciblée, mais aussi les brûlures au niveau des plaques dispersives. Un certain nombre de recommandations ont été faites par les spécialistes pour diminuer la fréquence et l'incidence de ces complications [RDC<sup>+</sup>04]. La cautérisation de la trajectoire de l'aiguille lors du retrait ou remplacement de l'aiguille permet de minimiser les risques d'hémorragie mais aussi de détruire les cellules cancéreuses qui auraient pu adhérer à l'aiguille. La localisation de la tumeur doit faire l'objet d'une attention particulière. En effet le placement de l'électrode peut être particulièrement délicat lorsque la tumeur est profonde et que la trajectoire pour l'atteindre est obstruée par de gros vaisseaux sanguins. L'imagerie per-opératoire a alors un rôle primordial pour assister une insertion évitant de perforer les vaisseaux si possible. Par ailleurs, lorsque la tumeur hépatique est sous-capsulaire<sup>3</sup>, à proximité d'un organe particulièrement sensible (par exemple le colon ou l'estomac), certaines précautions permettent de préserver ces organes des dommages dus à la zone de nécrose induite par le traitement. Il peut être décidé de déplacer l'organe concerné par rapport à la capsule grâce à des outils spécifiques ou tout simplement de diminuer la marge recommandée de tissus sains à détruire autour de la tumeur. De même cette marge peut être diminuée dans les cas de patients cirrhotiques qui présentent des risques élevés d'insuffisance hépatique ou lorsque la réserve hépatique risque d'être insuffisante pour que le foie assure ses fonctions<sup>4</sup>.

Afin de pouvoir proposer des outils de planification spécifiques à la radiofréquence, et de dégager les contraintes intervenant dans le choix d'une stratégie d'intervention, nous

---

2. par opposition aux opérations à visée palliative pour lesquelles la destruction complète des tissus cancéreux n'est pas envisagée à cause d'une expansion irrémédiable de la maladie

3. c'est-à-dire située à proximité de la capsule, l'enveloppe externe du foie

4. dans les cas de foie non cirrhotique, un minimum 30% du volume du foie doit être préservé

nous sommes intéressés aux approches des spécialistes, que ce soit en assistant à un certain nombre d'opérations, en les interrogeant, ou en parcourant la littérature médicale sur le sujet. Nous détaillerons les contraintes retenues dans le cadre de notre logiciel de planification dans la section 3.4.1. Nous allons maintenant évoquer, sans entrer dans les détails, les interventions similaires pour lesquelles notre travail est applicable.

## 1.2.2 Autres types d'ablation minimalement invasives

### 1.2.2.1 Cryoablation

Le principe de la cryoablation<sup>5</sup> est relativement similaire à la radiofréquence : le médecin insère une sonde spécifique dans les tissus à détruire (par voie percutanée, laparoscopique ou chirurgicale) et active le traitement. Au lieu de chauffer les tissus, la sonde de cryoablation permet de diminuer fortement leur température. L'argon circulant dans la sonde permet de geler les tissus à proximité jusqu'à atteindre des températures entre -20 et -100°C. Le mécanisme de destruction cellulaire par le froid est plus complexe que par la chaleur. En effet alors qu'il suffit d'atteindre une certaine température pendant un certain temps dans les tissus pour les détruire par chaleur, au-delà d'un certain seuil le froid peut au contraire permettre de conserver les tissus. La vitesse de refroidissement ainsi que la phase de dégel déterminent également la destruction ou conservation des cellules. Un cycle de cryoablation comporte deux phases de gel (approximativement 10 minutes chacune), séparées par une phase de dégel (5 minutes).

Le principal avantage de la cryoablation réside dans la possibilité de visualiser l'ellipsoïde de glace correspondant à la zone de nécrose quelle que soit la modalité d'imagerie accompagnant l'opération. De plus contrairement à la RFA, elle possède un effet anesthésiant intrinsèque qui permet de diminuer la douleur per- et post-opératoire. Elle permet également de protéger des zones ou organes sensibles grâce à des sondes chauffantes. Les points faibles de la technique sont la durée de la procédure (30 minute par cycle), et le coût des équipements plus élevés que ceux de la radiofréquence. Enfin des études comparatives entre les deux techniques semblent démontrer un léger avantage à la RFA en terme de taux de complications, mais aussi de récurrences locales des tumeurs traitées [AHL<sup>+</sup>02].

### 1.2.2.2 Alcoolisation

Contrairement aux autres techniques que nous venons de présenter, l'ablation de tumeur par alcoolisation ne consiste pas à détruire les cellules tumorales par effet thermique mais grâce à l'injection de plusieurs doses d'éthanol. L'ablation par alcoolisation a donc lieu en plusieurs séances (parfois plus d'une dizaine) et se déroule de façon assez similaire aux autres techniques d'ablation percutanée. Après une anesthésie locale, l'aiguille est enfoncée dans l'axe de la sonde d'échographie, puis suivie en temps réel jusqu'à ce qu'elle soit placée au centre de la tumeur. Le volume d'alcool préalablement déterminé est injecté sous contrôle échographique. La diffusion de l'alcool apparaît sous la forme d'une

---

5. également appelée cryothérapie ou cryochirurgie

image hyperéchogène. L'effet toxique de l'alcool sur la tumeur est multiple. Il détruit les protéines des cellules et bloque la microcirculation à l'intérieur de la tumeur. Ces deux effets combinés aboutissent à la nécrose des cellules ayant subi les injections. De plus la diffusion de l'alcool dans les cellules cancéreuses est favorisée par l'hypervascularisation des tumeurs par rapport aux tissus sains. Cependant cette diffusion ainsi que le degré de destruction des cellules touchées sont difficilement prévisibles, ce qui entraîne de multiples séances d'alcoolisation pour traiter une tumeur. Cette technique est assez efficace pour des petites tumeurs inférieures à 3 cm de diamètre. Au-delà de cette taille le contrôle de la zone traitée étant difficile, la subsistance de cellules tumorales après le traitement est fréquent. Les avantages de cette technique sont sa simplicité et son coût très faible.

### 1.2.2.3 Autres opérations de thermo-ablation

Il existe plusieurs autres techniques de thermo-ablation comparables à la RFA : l'ablation par laser, l'ablation par micro-ondes ainsi que l'ablation par ultrasons focalisés. Ces techniques sont à l'heure actuelle moins populaires et moins documentées que la RFA, mais leur existence montre tout de même l'intérêt croissant pour les techniques de thermo-ablation ces dernières années. Les deux premières techniques sont très proches de la RFA et diffèrent principalement par la source d'énergie utilisée pour provoquer l'échauffement des tissus.

L'ablation par micro-ondes utilise des micro-ondes électromagnétiques pour agiter les molécules d'eau dans la tumeur et dans les tissus adjacents. Cette vibration moléculaire entraîne une augmentation de la température et tue la cellule [KTN<sup>+</sup>07]. Cette technique est moins sensible à la variation de la composition des tissus et au phénomènes de carbonisation et de vaporisation, ce qui permet une augmentation de température plus importante dans les tissus et entraîne une durée d'ablation plus courte. Cette technique est très récente et donc peu étudiée à l'heure actuelle.

Les lasers produisent un rayonnement monochromatique. Il en existe une grande variété, de longueurs d'onde et de puissance différentes. La chaleur est produite par l'absorption de l'énergie lumineuse. La lumière laser est conduite par une fibre optique flexible, terminée par un diffuseur spécialement adapté. L'ablation par laser (Laser-Induced ThermoTherapy, LITT) consiste à plonger la fibre optique au coeur de la tumeur. La diode laser émettrice de lumière permet ensuite de brûler localement la tumeur.

L'intérêt de l'ablation par ultrasons focalisés de haute intensité (HIFU pour "High Intensity Focused Ultrasound") par rapport aux techniques nécessitant l'introduction d'une sonde réside dans le fait que les ultrasons peuvent être utilisés de manière totalement non invasive [GCB<sup>+</sup>00]. Les ultrasons sont des ondes sonores inaudibles qui se propagent parfaitement en milieu liquide, mais également dans les tissus biologiques. Ils sont communément utilisés pour les échographies, traversant les tissus sans que ceux-ci subissent aucun effet biologique, étant donné que les puissances utilisées sont très faibles. Lorsque le faisceau d'ultrasons est convergent et que les ondes ultrasonores de haute intensité sont focalisées en un point, l'effet immédiat obtenu au sein des tissus biologiques est leur échauffement brutal et extrêmement intense. Les températures atteintes sont telles qu'ils peuvent être

détruits par coagulation en quelques secondes. En déplaçant légèrement la sonde entre chaque impulsion et en répétant les tirs d’ultrasons, il est possible de détruire un volume conséquent. Cette technique est particulièrement adaptée dans les cas de cancer de la prostate. Une sonde produisant des ultrasons est introduite par voie rectale et la prostate est détruite dans son ensemble sans qu’une incision ne soit nécessaire.

### 1.3 Planification pré-opératoire assistée par ordinateur

Comme nous l’avons déjà évoqué dans le paragraphe 1.1, le couplage naturel des chirurgies mini-invasives avec l’imagerie, ainsi que les difficultés spécifiques entraînées par le manque de retour visuel et de dextérité inhérents à ces techniques ont poussé de nombreuses équipes de recherche à travailler sur l’assistance informatique ou robotique à ce type d’opérations. Nous nous intéressons ici en particulier à l’assistance pré-opératoire. Les travaux existants dans ce domaine ciblent principalement deux aspects de la planification pré-opératoire. Le premier aspect est le choix d’une stratégie adaptée à chaque cas, la puissance de calcul de l’ordinateur permettant d’étudier un certain nombre de possibilités et de proposer des solutions optimisées. Le second aspect est la prédiction des conséquences thérapeutiques correspondant à une stratégie opératoire. En effet dans les cas d’ablations de tumeurs, il est important de savoir quelle sera la zone effectivement traitée si l’on choisit telle ou telle stratégie. Cette information est assez délicate à avoir quelle que soit la technique, d’autant que comme nous avons pu le voir dans le paragraphe 1.2.1.3 la documentation sur l’efficacité des différents matériels existant est assez limitée et que de nouveaux matériels sont régulièrement développés.

#### 1.3.1 Simulation du traitement

Une bonne connaissance de la zone effectivement traitée est cruciale pour la réussite de l’opération. En effet, une sur-estimation de celle-ci entraînerait probablement la persistance de cellules cancéreuses après l’opération et par conséquent une récurrence de la tumeur dans les mois qui suivent l’opération. Un certain nombre d’expériences ont été menées<sup>6</sup> pour mesurer les nécroses effectives avec différents matériels, différentes techniques, différents paramètres. Cependant ces expériences sont coûteuses et ne permettent pas de moduler facilement les différents paramètres pouvant interférer avec la taille de la nécrose finale. La simulation théorique de cette nécrose grâce à des modèles physiques implémentés informatiquement semble alors être une approche efficace pour pouvoir étudier les effets des différents paramètres entrant en jeu, et notamment l’anatomie spécifique du patient lors de chaque opération.

Les approches de simulation de la nécrose induite par thermo-ablation consistent à déterminer l’évolution de la température au cours du temps dans un modèle plus ou moins complexe simulant le comportement de la zone traitée. Cette évolution peut être calculée

---

6. in vivo ou ex vivo dans des foies de porcs généralement



à une position donnée du modèle en tenant compte de la chaleur transférée par conductivité thermique et électrique ainsi que de la perfusion des tissus. Il est alors nécessaire de connaître les caractéristiques physiques des différents types de matériaux inclus dans le modèle. La résolution numérique des équations aux dérivées partielles décrivant l'évolution de la température au cours du temps en tout point du modèle passent par l'utilisation de la méthode des éléments finis. Cette méthode permet de résoudre de manière discrète le problème de la distribution de température dans le domaine défini par le modèle. Les études de modélisation considèrent alors généralement que la lésion résultante au traitement est équivalente à l'intérieur de la surface isotherme dont la température critique choisie est approximativement  $50^{\circ}\text{C}$  [TSH<sup>+</sup>02]. Pourtant comme nous l'avons évoqué à la section 1.2.1.1, la destruction cellulaire est à la fois conditionnée par la température et la durée d'exposition à la chaleur. L'ampleur de la nécrose peut alors être surestimée si l'on considère uniquement les températures atteintes dans le modèle, c'est pourquoi certains travaux prennent en considération le temps d'exposition dans la simulation [CN04].

Les études qui ont été menées dans ce domaine ont principalement pour objectif d'avoir une meilleure connaissance de l'impact des différents paramètres sur les dimensions et la forme de la nécrose à laquelle on peut s'attendre. Elles permettent notamment d'observer la déformation de cette forme due à la perte de chaleur liée à la perfusion des tissus qui est la source majeure de persistance de cellules cancéreuses après l'ablation d'une tumeur proche de gros vaisseaux sanguins [JW00, PWP05]. Ces simulations n'ont que peu été utilisées dans un objectif d'optimisation de placement d'outil. En effet, l'intérêt de la détermination de la nécrose par simulation d'échange thermique est, entre autres, de pouvoir tenir compte du contexte dans lequel on se place. Les processus d'optimisation, quelle que soit la technique choisie, nécessitent de tester différentes possibilités et donc d'effectuer la simulation à de nombreuses reprises avec des contextes différents, alors que la simulation des phénomènes physiques dans les tissus peuvent nécessiter de la puissance calculatoire plus ou moins importante selon la complexité du modèle. Quelques travaux de simulation ont tenté de combiner la simulation de la nécrose par des modèles physiques avec l'optimisation du placement d'aiguille. Chen *et al* [CMG06] ont présenté un modèle réduit à 2 dimensions ce qui implique à la fois un modèle par éléments finis réduit et une recherche d'optimum simplifiée par le fait que l'inconnue du problème est définie par seulement 3 paramètres, 2 correspondant au placement de l'origine de l'outil et 1 pour l'angle. Malgré cette simplification l'auteur reporte des temps d'exécution de l'ordre de 10 à 15 minutes. Des travaux comparables ont été présentés pour le cryochirurgie de la prostate [RLT04] avec des résultats équivalents en terme de temps d'exécution sur un modèle simplifié en 2 dimensions.

Nous avons donc décidé d'utiliser un modèle de la lésion plus simple pour notre processus de planification automatique. Notre modèle s'appuie sur les conclusions des différentes expérimentations qui ont été faites soit *in vivo* soit en simulation : la lésion de radiofréquence peut être assimilée à un sphéroïde dont le rapport entre les axes est fixe et dépend du matériel utilisé et dont la taille croît avec la puissance du traitement [DBDW<sup>+</sup>01]. Les praticiens considèrent généralement ce type de modèle, voire de simples sphères lors de

l'étape de planification pré-opératoire de l'opération. La détermination de la lésion par simulation des échanges thermiques n'est pas pour autant inutile à la phase de planification pré-opératoire, cependant il nous paraît plus approprié d'utiliser ces techniques après avoir sélectionné un nombre réduit de stratégies, afin de vérifier l'impact de celles-ci de manière plus précise et éventuellement de les départager. Dans l'article [JS02] Johnson *et al* proposent un modèle 3D de simulation de la température dans les tissus, dans l'objectif de contrôler la température atteinte pendant l'opération et d'adapter le traitement si certaines zones ciblées n'ont pas subi le dommage thermique nécessaire. Le temps d'exécution de la simulation doit alors être limité si l'on souhaite pouvoir fournir les informations de température de manière interactive pendant l'opération. Malgré cette contrainte forte de temps de calcul, le modèle proposé est résolu généralement en 1 à 2 minutes. Ce délai semble raisonnable tant que le nombre de simulations à effectuer est limité à 2 ou 3 solutions préselectionnées.

### 1.3.2 Aide à la décision

Dans le domaine de la chirurgie mini-invasive, l'apport des méthodes d'intelligence artificielle à la sélection d'une stratégie opératoire optimale n'a encore été que peu explorée. Pourtant les capacités calculatoires des machines actuelles devraient permettre de parcourir de nombreuses possibilités de stratégies et d'en extraire les plus intéressantes. A l'heure actuelle, le développement de ces techniques est encore limité par le fait qu'elles nécessitent d'avoir accès aux données patients de façon utilisable pour la machine, c'est-à-dire que la connaissance anatomique spécifique à chaque patient doit pouvoir être extraite facilement des images médicales. Cependant les progrès constants concernant à la fois les modalités d'imagerie médicale et les techniques de segmentation des images obtenues laissent présager un fort développement des recherches impliquées dans les étapes suivantes de la chirurgie assistée par ordinateur, c'est-à-dire la planification et la simulation. Généralement les travaux s'intéressant à la problématique de la planification automatisée du placement d'outils chirurgicaux se concentrent sur une intervention spécifique. Nous allons donc explorer les différents travaux proches de notre problématique trouvés dans la littérature du domaine médical, puis nous récapitulerons les problématiques qui semblent communes aux différentes applications.

#### 1.3.2.1 Cryochirurgie de la prostate

On peut citer les travaux portant sur le cryochirurgie de la prostate. La problématique dans le cas de la prostate n'est pas tout à fait la même que celle qui nous intéresse. Il s'agit de placer de multiples sondes de cryochirurgie de façon à détruire au maximum les tissus de la prostate tout en préservant à la fois les organes voisins et l'urètre qui le traverse [BSD<sup>+</sup>00]. Du point de vue de la planification le problème est assez complexe pour le praticien puisqu'il faut déterminer de façon optimale le nombre et la position des ablations nécessaires à une destruction quasi complète sans dommages externes. La fonction objectif à optimiser intègre alors généralement à la fois le critère de destruction des tissus ciblés et le



critère de préservation des autres tissus. L'axe d'insertion des sondes n'intervient pas dans l'optimisation puisque celles-ci sont introduites par voie rectale. L'article [LSR04] propose une heuristique d'optimisation basée sur les forces d'attraction et de répulsion exercée par les zones traitées incorrectement sur la position des différentes sondes de cryothérapies. Cette méthode permet de faire évoluer la stratégie de placement en limitant le nombre d'évaluations de la fonction objectif nécessaires à chaque étape du processus d'optimisation. Cette heuristique est spécifique à la fonction objectif concernée et ne pourrait être adaptée à d'autres objectifs d'optimisation.

### 1.3.2.2 Thermo-ablations de tumeurs hépatiques

Dans l'article [BWT<sup>+</sup>00], Butz *et al* présentent une approche ayant pour objectif la sélection d'une stratégie de placement d'aiguille optimale adaptable à différentes procédures d'ablation de tumeurs. Ce travail aborde déjà l'idée que les problématiques de détermination d'une trajectoire d'attaque pour différentes interventions de chirurgie mini-invasives sont relativement similaires et devraient pouvoir être traitées par des méthodes communes. Cependant la fonction objectif présentée prend en considération uniquement le fait que la zone de nécrose consécutive au placement de l'outil doit inclure un maximum de tissus cancéreux et un minimum de tissus sains. En réalité les paramètres intervenant dans le choix d'un placement sont multiples et beaucoup plus complexes. En premier lieu par exemple, une trajectoire permettant de détruire toute la tumeur en préservant au maximum les tissus sains n'aura pas lieu d'être si elle implique de perforer un organe vital à proximité alors qu'il existe des trajectoires plus sûres permettant de pratiquer l'intervention sans dommages. Altrogge *et al* ont également proposé des méthodes d'optimisation du placement d'aiguille dans le cas de la radiofréquence [AKP<sup>+</sup>06]. La fonction objectif présentée dépend de la distribution de la température dans les tissus et donc de la destruction des tissus ciblés. Cependant cette optimisation ne prend pas en considération le fait que la trajectoire soit reproductible en pratique ou que d'autres critères pourraient intervenir dans le choix du praticien.

### 1.3.2.3 Interventions minimalement invasives robotisées

La planification pré-opératoire automatisée du placement des outils chirurgicaux a également un grand intérêt dans le cas des interventions mini-invasives robotisées. Ces interventions nécessitent de planifier virtuellement le placement du robot et les points d'entrée des outils avant de transférer la planification à l'équipement robotique afin qu'elle soit appliquée en salle d'opération. Les contraintes à prendre en compte pour sélectionner une planification adaptée sont multiples : pas de collisions entre les bras du robot, bonnes conditions de dextérité et de visibilité, préservation de l'intuition du chirurgien en maintenant l'orientation relative des yeux et des mains [SDHT02]. Adhami *et al* ont présenté des travaux qui visent à coupler planification assistée du placement des outils et reproduction du geste planifié par un robot dans le cas d'interventions cardio-vasculaires [ACMB00]. Les différents critères pris en compte sont soit éliminatoires soit à optimiser. De plus ces

critères dépendent du type d'outil concerné par l'insertion. Le placement de l'endoscope ne subit pas les mêmes contraintes que celui des bras du robot. Dans ces travaux le choix d'un placement repose sur une sélection préalable de solutions en nombre limité, ce qui rend possible une optimisation exhaustive. Une approche similaire est présentée dans [KWOH04] qui propose de combiner algorithmes génétiques et méthode utilisant le gradient pour l'optimisation du placement des points d'entrée des outils robotiques.

#### 1.3.2.4 Problématiques communes aux différentes interventions

Les problématiques présentées dans les études portant sur la planification automatique du placement d'outil pour différents types d'intervention mini-invasives sont relativement similaires :

- Le placement d'outil recherché est caractérisé par 2 points : la position de son point d'insertion et la position de sa pointe. Selon les interventions, il se peut qu'on ne s'intéresse qu'au point d'insertion (chirurgie mini-invasive robotisée) ou à la position de la pointe (cryochirurgie de la prostate), l'autre point étant contraint (point d'insertion) ou variable durant l'opération (pointe d'outil mobile à l'intérieur du corps).
- Les *critères* ou *contraintes* énoncés par les experts permettant de réaliser un choix de placement sont de nature géométrique.
- Un *domaine de recherche* est défini pour les placements recherchés. Il peut être large (point d'insertion appartenant à la surface de la peau) ou limité (zone de la peau présélectionnée), continu ou discret.
- Le ou les critères de choix d'un placement sont traduits en une *fonction objectif*, afin de pouvoir appliquer des *méthodes d'optimisation*.
- Certains critères énoncés peuvent être éliminatoires pour un placement d'outil chirurgical. En particulier la question de l'évitement d'obstacles (organes à proximité du site chirurgical) est récurrente quelle que soit l'intervention considérée.

#### 1.3.2.5 Positionnement de ces travaux

La plupart des travaux existant dans le domaine se concentrent sur une opération particulière, pour laquelle les critères de choix du placement d'outil optimal sont souvent limités à un ou deux critères, ne tiennent pas compte des obstacles représentés par les organes vitaux du patient et proposent une heuristique spécifique à la fonction objectif étudiée. L'approche que nous avons voulu développer dans cette thèse est modulaire. Les contraintes intervenant dans le choix stratégique final du praticien sont formalisées dans un cadre précis qui peut intégrer un maximum de contraintes différentes. La résolution des contraintes est totalement indépendante de leur définition et s'adapte donc à n'importe quelles contraintes entrant dans le cadre défini. La modification de ces contraintes suite à une évolution des règles de planification de l'intervention est donc aisée. De plus ce travail, centré à l'origine sur l'ablation de tumeurs hépatiques par radiofréquence, doit pouvoir être réutilisé pour des interventions similaires moyennant des adaptations mineures. Les thérapies de thermo-ablation de tumeur entrent naturellement dans ce cadre, cependant

on pourrait adapter nos travaux à d'autres thérapies comme par exemple certaines interventions de neurochirurgie (placement d'électrodes de Stimulation Cérébrale Profonde, ablation de cavernomes).

Enfin un autre objectif fort de ce travail est de fournir un résultat satisfaisant en un temps de résolution réduit. En effet, ce travail consiste à proposer un plan de placement d'outil et de puissance du traitement en se basant sur les images pré-opératoires. Des images peuvent avoir été acquises plusieurs jours voire semaines avant l'intervention, cependant pendant ce temps le patient peut avoir grossi ou maigri, la tumeur évolué, mais surtout la position du patient dans le scanner peut ne pas correspondre à celle du jour de l'intervention, ce qui impliquerait un déplacement de tous ses organes internes par rapport au moment de l'intervention. Afin que la planification proposée puisse être valide, il est nécessaire qu'elle soit effectuée sur des images qui correspondent aux conditions de l'opération, acquises immédiatement avant celle-ci. Il n'est donc pas envisageable que l'outil de planification monopolise le patient, le praticien ainsi que la salle scanner dans laquelle se déroule l'opération plus de quelques minutes. Les méthodes de résolution et d'optimisation ainsi que le modèle de la nécrose induite par radiofréquence ont donc été étudiées en tenant compte de cette contrainte.



# Chapitre 2

## Contraintes et optimisation

### Sommaire

---

<b>2.1</b>	<b>Modélisation géométrique</b>	<b>29</b>
2.1.1	Modélisation : approche déclarative <i>vs</i> approche procédurale . .	29
2.1.2	Modélisation géométrique en CAO . . . . .	30
2.1.3	Modélisation par contraintes . . . . .	31
2.1.4	La modélisation déclarative . . . . .	32
2.1.4.1	Le processus de modélisation . . . . .	32
2.1.4.2	Modeleurs déclaratifs . . . . .	34
	La phase de description . . . . .	34
	La phase de résolution . . . . .	34
	La phase de prise de connaissance des solutions . . . . .	35
2.1.5	L'approche déclarative pour la planification d'opération mini-invasive	35
<b>2.2</b>	<b>Optimisation</b>	<b>36</b>
2.2.1	Définitions et terminologie . . . . .	36
2.2.2	Méthodes d'optimisation monocritère multidimensionnelles . .	37
2.2.2.1	Les méthodes d'optimisation locale . . . . .	38
	La descente du gradient . . . . .	38
	La méthode de Nelder-Mead (descente du simplexe) . . .	39
2.2.2.2	Les méta-heuristiques globales . . . . .	40
	La méthode tabou . . . . .	41
	Le recuit simulé . . . . .	41
	Les algorithmes évolutionnaires . . . . .	41
2.2.3	Choix d'une méthode d'optimisation . . . . .	42
<b>2.3</b>	<b>Les problèmes d'optimisation multi-objectifs</b>	<b>43</b>
2.3.1	Problématique . . . . .	43
2.3.2	Optimalité au sens de Pareto . . . . .	43
2.3.3	Méthodes de résolution muticritères . . . . .	44
2.3.3.1	Les méthodes agrégées . . . . .	45
	La méthode de pondération . . . . .	45
	La méthode $\epsilon$ -contrainte . . . . .	46

---

Programmation par but . . . . .	46
2.3.3.2 Méthodes traitant séparément les objectifs . . . . .	48
La méthode lexicographique : . . . . .	48
L'algorithme VEGA : . . . . .	48
2.3.3.3 Les méthodes Pareto . . . . .	48
2.3.4 Discussion . . . . .	49

---

LES travaux proposés dans cette thèse s’inspirent d’approches et techniques développées pour deux problématiques vastes en informatique : la satisfaction de contraintes en modélisation géométrique et les problèmes d’optimisation complexe. Nous considérons la problématique du choix d’une stratégie d’insertion d’outil de thermo-ablation comme un problème de modélisation à base de contraintes géométriques : le praticien expérimenté détermine les propriétés d’une insertion adéquate, le solveur propose les solutions correspondantes. D’autre part, l’assistance à la planification du praticien consiste à proposer la *meilleure* stratégie opératoire *possible* pour chaque patient subissant une opération de thermo-ablation. Le terme *meilleur* implique une problématique d’optimisation, et le terme *possible* sous-entend qu’un certain nombre de contraintes restreignent le domaine de recherche. Les problèmes d’optimisation sous contraintes sont classiques et touchent de nombreux domaines, en informatique, en économie, etc. En réalité, nous sommes confrontés en permanence à des problèmes d’optimisation pour des choix concrets, pour lesquels la plupart du temps plusieurs critères entrent en ligne de compte. Pour le choix d’une voiture par exemple : le prix, la consommation, la sécurité, l’esthétique font partie des critères intervenant dans la décision finale. Certains d’entre eux ne sont pas forcément concordants. Il s’agit alors de sélectionner la ou les voitures correspondant au meilleur compromis pour ces différents critères. Il en est de même pour le choix d’un placement d’outil chirurgical pour lequel plusieurs critères sont à considérer.

Dans un premier temps nous allons présenter les approches de la modélisation géométrique dont nous nous inspirons notamment dans le domaine de la conception assistée par ordinateur. Ensuite nous introduirons la problématique de l’optimisation sous contraintes. Il n’existe pas de méthode universelle permettant de répondre à cette problématique difficile dans tous les cas. Nous présenterons brièvement quelques-unes des approches et méthodes existantes. Ensuite nous décrirons le cas plus complexe de l’optimisation multi-objectifs pour laquelle le concept de solution optimale a été redéfini et les méthodes d’optimisations traditionnelles ont été adaptées.

## 2.1 Modélisation géométrique

### 2.1.1 Modélisation : approche déclarative *vs* approche procédurale

Dans le processus de modélisation, le rôle de l’ordinateur est d’effectuer les calculs nécessaires pour simuler le comportement du système à modéliser. Les modèles sont alors naturellement construits comme des programmes informatiques. Cette approche de la modélisation est procédurale puisque les programmes sont constitués d’une suite de procédures à exécuter. A cette approche s’oppose l’approche déclarative, qui est basée sur le principe qu’un modèle doit être représenté par une spécification qui décrit les objets, les variables du modèle et les relations fonctionnelles entre elles. La spécification du modèle est alors séparée de l’outil développé pour effectuer la conception du modèle. L’approche procédurale consiste donc à donner des instructions (= programme) permettant de dire *comment* construire le modèle, alors que l’approche déclarative consiste à définir *quel* est le modèle

(= données). Prenons un exemple pour illustrer cette distinction et supposons que l'on veuille aider quelqu'un à aller à un endroit précis. On a alors la possibilité de lui expliquer comment y aller (tournez à droite, marchez 100 mètres, etc), ou alors on peut lui donner une carte en se basant sur le fait que le cerveau pourra à partir de ces données appliquer les instructions qui correspondent.

On peut représenter la distinction entre ces deux approches par une frontière entre les informations représentées déclarativement et le programme exploitant ces informations. Cette frontière entre les données et le programme n'est pas fixe. Quel que soit le problème on peut décider ce qui doit être représenté par des données et ce qui doit être programmé dans un langage procédural. Plus la quantité d'information représentée par des données est importante, plus le programme qui les interprète devra être générique. Les avantages à déplacer cette frontière du côté déclaratif sont multiples et sont résumés dans la table 2.1. Notons tout de même que le principal inconvénient de passer des informations par approche déclarative implique souvent une perte d'efficacité due à la généralité du programme. En effet, l'approche procédurale permet d'optimiser les calculs pour l'utilisation spécifique concernée alors que l'approche déclarative implique que l'on doit pouvoir s'adapter à tout type de données.

TABLE 2.1: Comparaison approche procédurale / approche déclarative

Approche déclarative	Approche procédurale
+ réutilisation des connaissances possible	- connaissance utilisée dans un seul but
+ facilement modifiable	- difficilement modifiable
+ compréhension intuitive	- compréhension difficile
+ collection indépendante de faits	- décomposition délicate
+ facile à valider	- difficile à corriger
+ transparent	- boîte noire
- efficacité moindre	+ grande efficacité

### 2.1.2 Modélisation géométrique en CAO

Dans le domaine de la conception assistée par ordinateur (CAO), les modèles sont représentés de manière *déclarative* dans le sens où chaque modèle est représenté par des données indépendantes du programme qui l'interprète. Ainsi ces données peuvent être utilisées par différents programmes pour une visualisation simple, en vue d'une édition, ou également pour d'autres tâches qu'on voudrait pouvoir réaliser pour chaque modèle géométrique. Il existe cependant différents modèles de représentation des connaissances permettant de définir partiellement ou totalement un modèle : esquisses cotées [MT09], modèles mathématiques, langage naturel [CS01][LG05], etc. Du point de vue de la machine un modèle géométrique peut être considéré comme un ensemble de points avec des coordonnées, reliés



entre eux par des arêtes et des faces. Cependant, du point de vue du concepteur, le modèle n'est pas une géométrie fixe et complète qu'il aurait directement en tête avant même le commencement de la conception. Le processus de conception peut être vu comme un processus de satisfaction de contraintes [MD04]. Il consiste en un enrichissement progressif de différents modèles de représentation des connaissances (mots du langage naturel, esquisses, modèles réduits, modèles mathématiques). L'enrichissement de ces modèles est réalisé en tenant compte des propriétés, caractéristiques et contraintes devant être satisfaites par l'objet à concevoir. Dans notre contexte, tout cela se traduit par des contraintes géométriques. La plupart des modélisateurs existant en CAO ne tiennent pas compte de la nature prospective du processus de conception, c'est donc généralement au concepteur de s'assurer que les contraintes régissant son modèle sont effectivement respectées tout au long de la modélisation. L'approche de la modélisation géométrique par contraintes permet de soulager le concepteur de cette tâche et de faciliter la description et les modifications du modèle.

### 2.1.3 Modélisation par contraintes

**Définition 1** Une contrainte est une expression logique reliant des variables de décision, chacune d'elles prenant une valeur dans un domaine.

*exemples* :  $x < y$ ,  $A \neq B$ ,  $C \in \{\text{blanc}; \text{noir}\}$

Une contrainte provoque une restriction sur les valeurs que peuvent prendre simultanément les variables du problème. A partir d'un ensemble de contraintes, un solveur de contraintes est capable de proposer un ou plusieurs modèles assurant le respect de chacune d'entre elles.

**Définition 2** Une solution à un problème  $P$  constitué de plusieurs contraintes est un ensemble d'instanciations des variables de décision (un  $n$ -uplet de valeurs) tel que toutes les contraintes sont satisfaites.

Selon le nombre de solutions correspondant à un problème  $P$  représenté par un ensemble de contraintes, celui-ci est dit :

- *sous-contraint* s'il existe plusieurs ou une infinité de solutions à  $P$ .
- *bien contraint* s'il existe une seule solution à  $P$ .
- *sur-contraint* s'il n'existe pas de solution à  $P$ .

Les avantages de la modélisation par contraintes sont multiples. Les relations entre les objets sont plus faciles à manipuler et à maintenir qu'un ensemble de fragments de code. La définition des contraintes du modèle permet au concepteur de se concentrer sur les relations qui doivent être maintenues dans le modèle plutôt que sur la façon de les maintenir. Le solveur de contraintes peut informer sur les relations satisfaites par le système et les raisons qui empêchent de les respecter le cas échéant.

L'utilisation des contraintes pour la modélisation implique également plusieurs défis pour l'implémentation des solveurs :

- La généralité : les solveurs devraient pouvoir interpréter différentes sortes de contraintes (par ex : linéaires ou non, égalité/inégalité, etc) de façon à ne pas restreindre son champ d'application.
- L'efficacité : le solveur doit fournir une solution même dans le cas d'un problème mal posé. En effet il se pose en particulier la question de la gestion des problèmes sous-contraints et sur-contraints. Dans le cas de problèmes sur-contraints il s'agit d'avoir la possibilité de relâcher un certain nombre de contraintes pour fournir une solution. La résolution de problèmes sous-contraints pose la question du calcul exhaustif des solutions (ou de leur représentation dans le cas d'un continuum), ou de la sélection d'une ou plusieurs solutions.
- La réactivité : le solveur doit être un système interactif permettant une évolution des relations entre les objets. Le processus de conception étant constitué d'un certain nombre de tentatives et d'ajustements, il doit être possible de retirer ou d'ajouter facilement des contraintes. Le solveur doit alors tirer parti de la moindre opportunité d'augmentation des performances pour mettre à jour les solutions de façon dynamique et rapide.

Kwaiter *et al* détaillent les différentes approches existantes dans le domaine de la modélisation par contraintes et exposent leurs avantages et leurs limites dans [KGC98].

## 2.1.4 La modélisation déclarative

### 2.1.4.1 Le processus de modélisation

L'approche déclarative pour la modélisation repose également sur la définition de contraintes. A la différence des approches classiques de modélisation par contraintes, la modélisation déclarative est plus proche du concepteur, elle propose des outils de conception de haut niveau. En effet la principale particularité d'un modèleur déclaratif est de pouvoir interpréter des propriétés décrivant les objets à modéliser dans un langage plus proche de celui du concepteur que de celui de la machine. Cette approche permet alors de rendre l'étape de conception très intuitive pour le concepteur [CDMM98]. De plus les propriétés décrivant la scène peuvent être imprécises, le langage naturel permettant de nuancer les propriétés. Par exemple, un objet peut être décrit comme "assez grand", une ligne comme "légèrement courbée", deux éléments de la scène comme "très proches", etc. Il n'est alors pas nécessaire d'avoir une idée précise du résultat attendu au début du processus de conception.

La figure 2.1 représente le processus de conception assisté par un modèleur déclaratif comparativement à l'utilisation d'un modèleur classique. Avec un modèleur classique (figure 2.1(a)), le concepteur part de l'idée initiale pour définir une spécification de l'objet à concevoir. Il imagine alors un objet conforme à cette spécification. Le modèleur fournit des outils pour faciliter la modélisation concrète de l'objet imaginé. Le modèleur permet de visualiser facilement l'objet conçu sous différents angles et de l'évaluer en fonction de l'idée et des spécifications initiales. Le concepteur peut alors valider le modèle ou adapter les spécifications du modèle si nécessaire puis imaginer un nouvel objet plus conforme à ses attentes. Dans ce cas le modèleur n'assiste pas réellement le concepteur dans le

processus de conception, il fournit simplement un support pour réaliser un objet précis. Le modelleur déclaratif intervient pleinement dans le cycle de conception (figure 2.1(b)). Le concepteur fournit une description déclarative dans un langage spécifique, le modelleur crée des solutions correspondant aux propriétés décrites, et permet alors au concepteur de naviguer dans les solutions de façon soit à en sélectionner une, soit à modifier ou raffiner sa description.

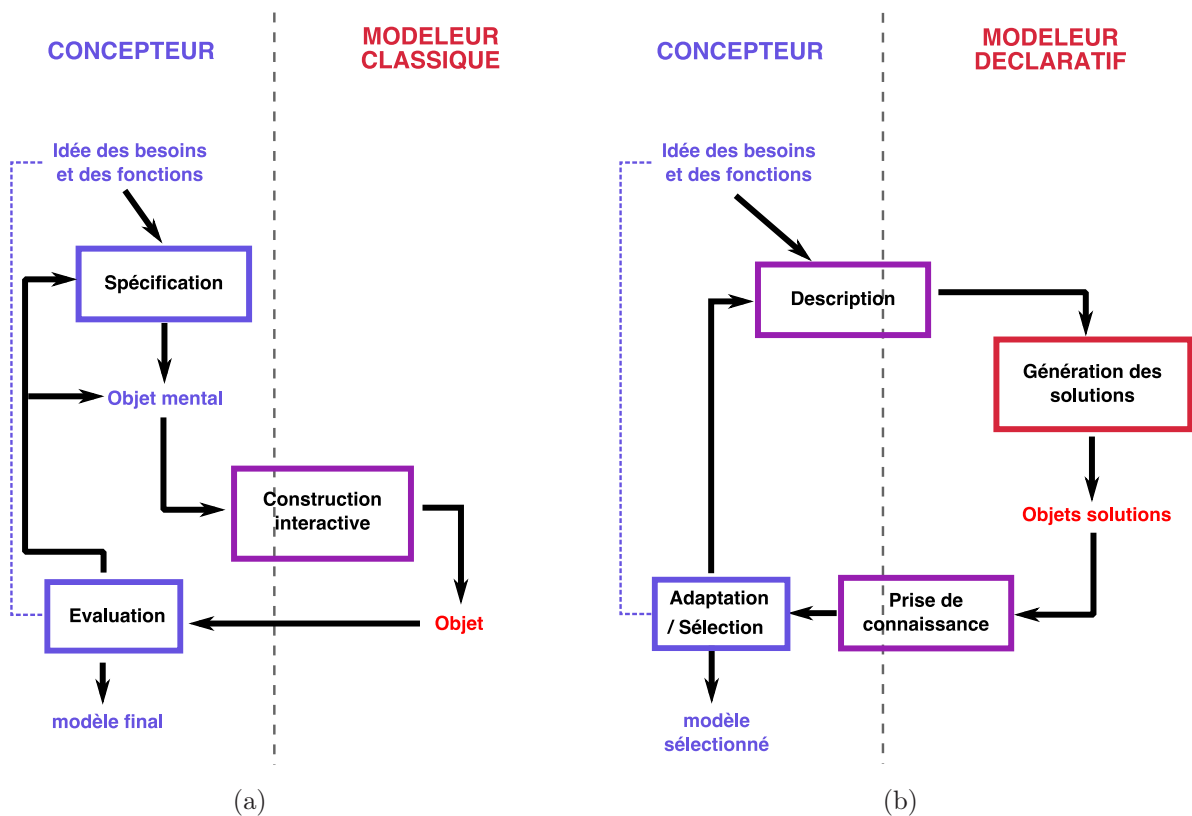


FIGURE 2.1: Comparaison du processus de conception avec un modelleur classique et un modelleur déclaratif

Les intérêts de la modélisation déclarative pour le concepteur de scène sont donc les suivants :

- la description est aisée et rapide : l'objet est défini par des propriétés et des contraintes de haut-niveau.
- les solutions sont toutes correctes et cohérentes avec les requêtes du concepteur, il est inutile d'en vérifier la validité.
- plusieurs objets peuvent être générés à la fois, permettant d'explorer différentes possibilités alors que la production d'un modèle par la voie traditionnelle est complexe et longue, limitant ainsi le nombre de cas étudiés.
- la facilité et la rapidité de production de solution permet de multiplier les essais et de concevoir de manière itérative le modèle tout en ayant un retour concret sur l'évolution du résultat.

### 2.1.4.2 Modeleurs déclaratifs

L'approche déclarative pour la modélisation a été principalement explorée dans le domaine de la CAO. Les modeleurs déclaratifs existants permettent par exemple la génération de polyèdres, la modélisation de bâtiments, la création d'environnements virtuels en se concentrant principalement sur la définition de propriétés géométriques (pour plus de détail voir [GP00, BPR04, Gai03]). Dans [CN05], Christie *et al* proposent une approche déclarative pour la problématique du placement de caméra virtuelle dans un environnement 3D. La vue résultante de la scène respecte un ensemble de propriétés visuelles naturellement décrites grâce au vocabulaire cinématographique spécifique. La problématique est proche de celle du placement d'outil chirurgical, avec notamment les questions d'occlusion / intersection.

Les modeleurs déclaratifs sont développés suivant trois axes, correspondant aux phases de conception [Gai07] :

- la phase de description : l'utilisateur décrit ses attentes grâce à des outils d'interaction ou un langage spécifique.
- la phase de résolution : le solveur produit un ou plusieurs modèles répondant à la description.
- la phase de prise de connaissance : le résultat de la résolution est présenté au concepteur. Celui-ci doit alors avoir la possibilité d'explorer les solutions et de raffiner sa description si nécessaire.

Nous allons donc présenter les problématiques impliquées dans chacune de ces phases de la conception.

**La phase de description** La phase de description permet l'énoncé des propriétés à l'aide d'une interface adaptée. C'est la partie visible pour le concepteur qui doit exprimer les propriétés de son modèle. L'énoncé et la modification des propriétés doivent être faciles et intuitifs. Cette phase peut être multimodale à travers une interface utilisant conjointement des périphériques graphiques, textuels, gestuels ou oraux. La description peut être effectuée de façon globale (définition d'un script décrivant totalement le modèle) ou incrémentale (définition enrichie au fur et à mesure du processus de conception). Le modeleur déclaratif comporte donc un module qui est chargé d'interpréter les propriétés définies par le concepteur et de les transformer en relations utilisables par le solveur de contraintes. L'interpréteur peut également se charger de vérifier que les propriétés énoncées sont cohérentes.

**La phase de résolution** Le solveur est le cœur du modeleur déclaratif. Il se charge de construire des solutions conformes à la description fournie. La résolution peut aboutir à la proposition d'une solution, de plusieurs solutions, de toutes les solutions. Dans le cas où il n'y a pas de solution au problème tel qu'il a été défini, certaines contraintes peuvent être relâchées de façon à ce que le modeleur assure la proposition d'une solution même dégénérée.

Le solveur utilise des méthodes de recherche opérationnelle et d'intelligence artificielle pour parcourir le domaine de recherche de façon efficace. Les approches possibles pour la phase de résolution sont les suivantes :

- les approches procédurales spécifiques à une application particulière. Elles ont l'avantage d'être efficaces, mais sont difficiles à adapter à d'autres contextes ;
- les approches déductives basées sur un système à base de règles ou un système expert ;
- les approches de parcours d'arbres qui permettent exploration systématique de l'espace de recherche [LRGC04]. Elles nécessitent cependant de discrétiser les domaines de définition des différentes variables ;
- les approches stochastiques qui reposent sur l'utilisation de méta-heuristiques. Ce sont des méthodes puissantes et flexibles. Elles permettent de traiter des espaces de recherche très vastes.

**La phase de prise de connaissance des solutions** L'approche déclarative permet au concepteur d'avoir à l'origine une idée très vague de son modèle final. Cela implique généralement qu'au début du processus de conception, les modèles correspondant à cette première idée sont multiples et donc que l'énoncé des contraintes original décrit un problème sous-contraint. Si le solveur permet de calculer de nombreuses voire toutes les solutions du problème, il ne peut en revanche pas les présenter de façon exhaustive au concepteur. Il est alors nécessaire de développer des techniques d'exploration de l'espace des solutions. Il peut s'agir de sélectionner certaines solutions [EV01] ou de les classer en regroupant des solutions proches au regard d'une métrique adaptée [Cha98].

De plus l'interface du modéleur doit permettre d'accéder facilement aux informations relatives aux modèles présentés, et peut intégrer des fonctionnalités de présentation des solutions générées [BDP99] de façon à en faciliter la compréhension et l'évaluation. Enfin l'interface doit également proposer différents outils pour permettre au concepteur de raffiner sa description et de sélectionner les modèles ou parties de modèle qui lui semblent satisfaisants.

### 2.1.5 L'approche déclarative pour la planification d'opération mini-invasive

Dans ces travaux, nous avons choisi d'aborder le problème de la planification d'une opération de chirurgie mini-invasive de façon similaire à certaines approches de modélisation par contraintes en CAO. Avant de pratiquer une opération de radiofréquence, le praticien n'a pas une idée précise de la façon dont il va aborder l'ablation de la tumeur, il se base sur un certain nombre de contraintes dictées par les règles de l'art ou par son expérience propre, pour choisir une stratégie qui lui paraît adaptée au cas considéré. Cependant de cette manière il est possible que celui-ci n'entrevoit qu'une partie des possibilités qui s'offrent à lui et ne sélectionne pas le meilleur choix correspondant aux contraintes entrant en considération, en particulier si son expérience de ce type d'opération est encore limitée. Nous nous rapprochons donc dans notre démarche des travaux existants en modé-

lisation par contraintes et en modélisation déclarative. Nous voulons proposer un solveur de contraintes assistant la phase de planification des opérations de chirurgie mini-invasive en sélectionnant des stratégies solutions correspondant à des contraintes définies de façon indépendante au solveur par un praticien expérimenté. Pour cela les travaux développés sont décomposés en trois axes correspondant au trois phases de la conception à l'aide d'un modeleur déclaratif. Il s'agira donc de proposer un langage de description de contraintes spécifique, de développer des techniques de résolution efficaces permettant de résoudre toutes les contraintes envisageables et enfin de proposer une interface utilisateur intuitive permettant de visualiser et parcourir les différentes stratégies possibles et éventuellement d'ajuster les contraintes de la planification.

## 2.2 Optimisation

La résolution des contraintes de la planification chirurgicale implique une problématique d'optimisation sous contraintes. Parmi les éléments appartenant à l'espace de recherche définissant les stratégies opératoires, certains sont des *solutions envisageables* dans le sens où elles obéissent à toutes les contraintes booléennes, d'autres sont des *solutions optimales* car elles satisfont au mieux un certain nombre de critères de sélections également définis dans l'ensemble des contraintes. La problématique de l'optimisation d'une fonction quelconque dans un domaine vaste et continu n'est pas simple et a été largement étudiée (cf [DPST03] pour un tour d'horizon). Nous nous sommes donc intéressés aux différentes méthodes permettant de traiter les problèmes classiques d'optimisation. Nous allons dans un premier temps définir ce qu'est un problème d'optimisation et les termes classiques qui s'y rapportent puis nous évoquerons quelques méthodes d'optimisation parmi les plus courantes, en particulier la méthode de descente du simplexe (également appelée méthode de Nelder-Mead) que nous utiliserons par la suite.

### 2.2.1 Définitions et terminologie

Un *problème d'optimisation* se définit comme la recherche du minimum ou du maximum d'une fonction donnée. Dans certains cas, les variables de la fonction à optimiser sont contraintes d'évoluer dans une partie de l'espace de recherche, on parle alors d'optimisation sous contrainte. Un problème d'optimisation peut être décrit mathématiquement de la façon suivante :

$$\text{chercher } x_0 \in D \quad \text{tel que} \quad f(\vec{x}_0) < f(\vec{x}), \quad \forall x \in D \quad (2.1)$$

avec

$$\begin{aligned} D \text{ défini par } & \vec{g}(\vec{x}_D) \leq 0, \quad \vec{h}(\vec{x}_D) = 0, \\ & \vec{x} \in R^n, \quad \vec{g}(\vec{x}) \in R^m, \quad \vec{h}(\vec{x}) \in R^p. \end{aligned}$$

Les vecteurs  $\vec{g}(\vec{x})$  et  $\vec{h}(\vec{x})$  représentent respectivement les contraintes d'inégalité et d'égalité qui délimitent un espace restreint de recherche de la solution optimale.

La fonction  $f$  est appelée *fonction objectif* (syn. *fonction de coût* ou *critère d'optimisation*). Elle représente le but à atteindre pour le décideur. On remarque que dans la définition mathématique d'un problème d'optimisation cette fonction est à minimiser. Lorsque l'on cherche à maximiser une fonction objectif, il suffit de transformer le problème en problème de minimisation dans lequel la nouvelle fonction objectif est  $-f$ . On considère alors tout problème d'optimisation comme équivalent à un problème de minimisation.

Les *variables de décision* du problème sont regroupées dans le vecteur  $\vec{x}$ . Lors du processus d'optimisation on cherche à faire varier ce vecteur de façon à déterminer la valeur des variables pour lesquelles la fonction objectif est minimale.

L'*espace de recherche* (syn. *espace d'états*) est défini par l'ensemble des domaines de définition des variables du problème. L'ensemble des *contraintes* (syn. *conditions*) définit les conditions sur l'espace d'états que les variables doivent satisfaire. Ces contraintes sont souvent des contraintes d'inégalité ou d'égalité et permettent en général de limiter l'espace de recherche. L'espace de recherche soumis aux contraintes est alors appelé *espace des solutions réalisables*.

Une *méthode d'optimisation* recherche le point ou un ensemble de points de l'espace d'état qui satisfait au mieux un (ou plusieurs) critère(s). Le résultat est appelé *valeur optimale* ou *optimum*.

$\vec{x}^*$  est un *minimum global* de la fonction  $f$  si  $\forall \vec{x} \in R^n$  tel que  $\vec{x} \neq \vec{x}^*$ , on a  $f(\vec{x}^*) < f(\vec{x})$ .

$\vec{x}^*$  est un *minimum local* de la fonction  $f$  si  $\forall \vec{x} \in V(\vec{x}^*)$  tel que  $\vec{x} \neq \vec{x}^*$ , où  $V(\vec{x}^*)$  définit un voisinage de  $(\vec{x}^*)$  on a  $f(\vec{x}^*) < f(\vec{x})$ .

## 2.2.2 Méthodes d'optimisation monocritère multidimensionnelles

Les problèmes d'optimisation peuvent être classés en deux grandes catégories : les problèmes combinatoires (c'est-à-dire pour lesquels les variables de décision évoluent dans un espace discret et fini) et les problèmes à variables continues. On s'intéresse dans cette partie uniquement aux méthodes d'optimisation concernant ce dernier type de problème même si certaines d'entre elles peuvent également être appliquées aux problèmes combinatoires. Pour ce type de problème il n'existe actuellement aucune méthode générique permettant d'aboutir à coup sûr au minimum global de la fonction objectif à moins que celle-ci ne soit particulièrement simple. On peut classer les méthodes d'optimisation continue selon plusieurs approches (voir figure 2.2) :

- les méthodes d'**optimisation locale** qui permettent de trouver de manière efficace un optimum local d'une fonction.
- les **méta-heuristiques globales** qui proposent des stratégies permettant d'éviter de converger vers un optimum local. Cependant, elles ne garantissent pas la découverte de l'optimum global en un temps fini.
- les **méthodes hybrides** qui tentent de tirer profit des avantages des deux précédentes approches.



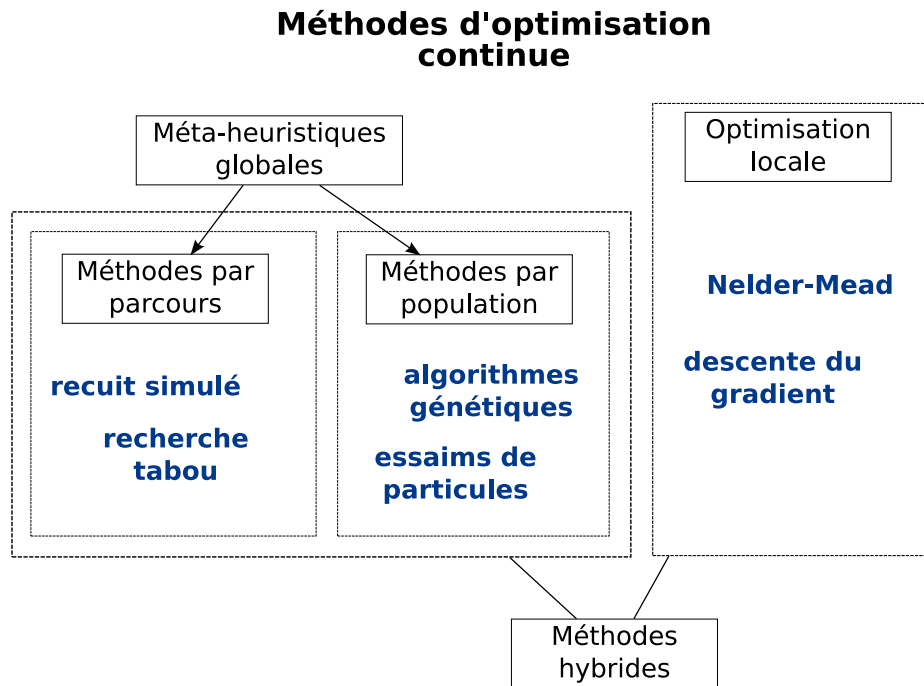


FIGURE 2.2: Classification de quelques méthodes d'optimisation dans le cas de variables continues

### 2.2.2.1 Les méthodes d'optimisation locale

Une première approche d'optimisation consiste à se déplacer d'une solution initiale vers une solution meilleure de manière itérative de façon à trouver un minimum local de la fonction. On peut alors lancer le processus un certain nombre de fois à partir de différents points initiaux de façon à augmenter les chances que le minimum local auquel on aboutit soit le minimum global recherché. Nous présentons ici brièvement deux d'entre elles, pour plus de détail sur les méthodes classiques d'optimisation locales, leurs avantages et leur mise en pratique nous invitons le lecteur à se référer à [PTVF02].

**La descente du gradient** La descente du gradient est basée sur le fait qu'une fonction  $F(\vec{x})$  différentiable en un point  $\vec{x}_0$  décroît le plus rapidement dans une direction opposée à celle du gradient de  $F$  en  $\vec{x}_0$ . On définit alors la suite  $x_n$  de la façon suivante :

$$\vec{x}_{n+1} = \vec{x}_n - \gamma_n \cdot \nabla F(\vec{x}_n)$$

Pour un pas  $\gamma_n$  suffisamment petit, on a  $F(\vec{x}_{n+1}) < F(\vec{x}_n)$ . La suite  $\vec{x}_n$  converge vers un minimum local de la fonction  $F$ . Cette méthode est relativement simple, cependant elle présente plusieurs inconvénients. Elle peut nécessiter de nombreuses itérations pour converger vers un minimum local et la recherche du pas optimal à chaque itération peut se révéler très longue. D'autres méthodes basées sur le gradient (méthode de Newton-Raphson par exemple) permettent une convergence plus rapide. Ces méthodes nécessitent de pouvoir calculer le gradient de la fonction à optimiser ce qui n'est pas toujours le cas.



**Algorithme 2.1** Nelder-Mead

---

```

1: Simplexe  $S$  constitué de  $N + 1$  points  $P_i$ 
2: tant que  $f(P_N) - f(P_0) > \epsilon$  faire
3:    $P_0 \leftarrow p \in S$  tel que  $f(p) = \min(f(P_i))$ 
4:    $P_N \leftarrow p \in S$  tel que  $f(p) = \max(f(P_i))$ 
5:    $G \leftarrow$  centre de gravité de  $\{P_0, \dots, P_{N-1}\}$ 
6:    $P'_N = G + (G - P_N)$  (Réflexion)
7:   si  $f(P'_N) < f(P_0)$  alors
8:      $P''_N = G + 2.(G - P_N)$  (Etirement)
9:     si  $f(P''_N) < f(P'_N)$  alors
10:       $P_N \leftarrow P''_N$ 
11:     sinon
12:       $P_N \leftarrow P'_N$ 
13:     fin si
14:   sinon
15:     si  $f(P'_N) > f(P_N)$  alors
16:        $P_N \leftarrow P_N + 1/2.(G - P_N)$  (Contraction simple)
17:     sinon
18:        $\forall P_i \neq P_0 \in S, P_i \leftarrow P_0 + 1/2.(P_i - P_0)$  (Contraction multiple)
19:     fin si
20:   fin si
21: fin tant que
22: Renvoie  $P_0$ 

```

---

**La méthode de Nelder-Mead (descente du simplexe)** Cette méthode locale effectue une recherche mutidirectionnelle dans l'espace de recherche. L'intérêt principal de la méthode du simplexe est qu'elle ne nécessite pas de calcul de gradient, elle est uniquement basée sur l'évaluation de la fonction objectif. Cette méthode est donc applicable à n'importe quel type de fonctions. Le second avantage de Nelder-Mead est d'être une méthode qui converge rapidement et précisément vers un minimum local. Ces caractéristiques nous ont poussés à utiliser cette méthode pour notre problème d'optimisation.

L'algorithme 2.1 présente le fonctionnement détaillé de la méthode. Cette méthode utilise le concept de simplexe qui est un ensemble de  $N + 1$  sommets formant un repère dans un espace à  $N$  dimensions. Le déroulement de l'algorithme sur un exemple simple en deux dimensions est illustré par la figure 2.3. Le principe de cette méthode est de faire évoluer au fur et à mesure les points du simplexe via des mécanismes de réflexion (fig. 2.3(b)), d'étirement (fig. 2.3(c)) et de contraction (fig. 2.3(f) et 2.3(h)) du simplexe, de façon à améliorer les valeurs de la fonction objectif correspondant aux points du simplexe. L'algorithme se termine et propose un résultat lorsque la différence d'évaluation entre le meilleur et le pire point du simplexe est négligeable.

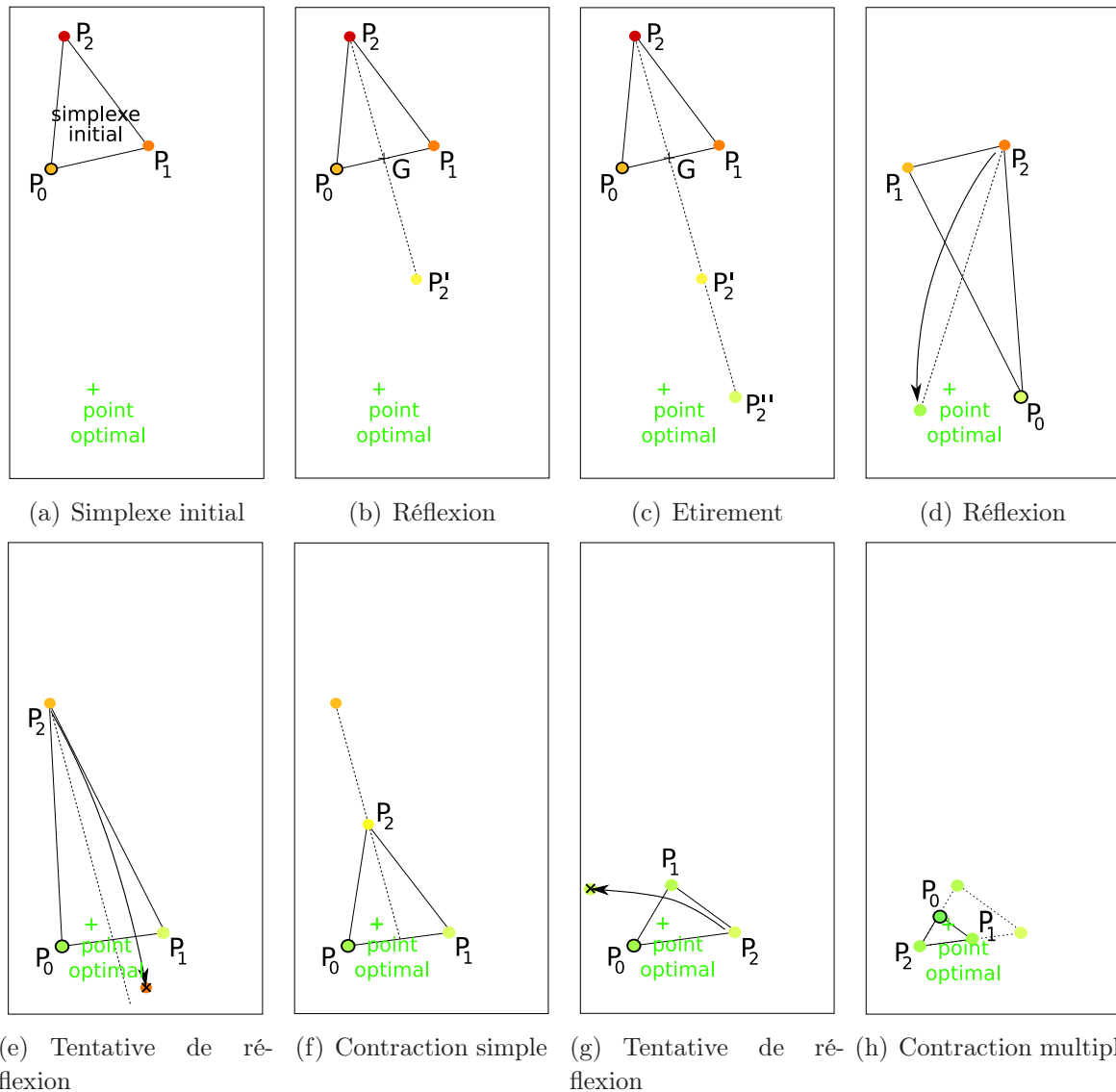


FIGURE 2.3: **Méthode de Nelder-Mead, exemple en dimension 2** : Le simplexe est représenté par un triangle. L'évaluation de la fonction objectif pour chaque point est représenté par des couleurs, du vert pour les valeurs faibles au rouge pour les valeurs fortes. Les points du simplexe sont classé dans l'ordre croissant de leur évaluation. A la dernière étape le point  $P_0$ , correspondant à l'évaluation la plus favorable du simplexe est proposé comme résultat de l'optimisation.

### 2.2.2.2 Les méta-heuristiques globales

La difficulté principale d'un problème d'optimisation est d'éviter de converger trop rapidement vers un minimum local de la fonction objectif. Différentes heuristiques ont été développées, proposant des stratégies permettant de s'extraire des minima locaux. Nous présentons ici quelques unes des plus populaires. Nous présentons tout d'abord la méthode tabou et le recuit simulé qui sont des méthodes d'optimisation par exploration du voisinage à partir d'une solution initiale. Puis nous évoquerons les algorithmes évolutionnaires qui fonctionnent par manipulation d'une population de solutions.

**La méthode tabou** Le fonctionnement de la méthode tabou est proche de certaines méthodes de recherche locale comme la descente du gradient. Le principe de ces méthodes locales est de partir d'une position initiale, et d'explorer itérativement le voisinage de la position courante afin de choisir une nouvelle position, pour converger finalement vers un minimum. Dans le cas de la méthode tabou, il s'agit de sélectionner la position du voisinage qui correspond à la meilleure évaluation de la fonction objectif. Lorsque la position courante est un minimum, il n'est pas possible de trouver une meilleure position dans le voisinage. La meilleure solution du voisinage devient la position courante, malgré le fait qu'elle entraîne une augmentation de la fonction objectif. Grâce à ce mécanisme, la recherche n'est pas piégée par les minima locaux. Cependant si l'on considère l'itération suivante, la meilleure solution du voisinage est probablement le minimum auquel on vient d'échapper. Pour éviter que la recherche boucle indéfiniment comme dans l'exemple décrit, une liste *tabou* a été introduite regroupant les solutions précédemment visitées. A chaque itération, la position courante est alors sélectionnée parmi les solutions n'appartenant pas à cette liste de solutions interdites.

**Le recuit simulé** Le recuit simulé est une métaheuristique inspirée d'un procédé métallurgique qui consiste à réchauffer puis refroidir lentement un métal [Kir84]. Le réchauffement pousse les atomes à se déplacer aléatoirement par rapport à leur position initiale qui correspond à un minimum local de l'énergie interne du matériaux. Le refroidissement lent permet aux atomes de retrouver une configuration pour laquelle l'énergie interne résultante est potentiellement plus intéressante que la configuration initiale. Le principe de la méthode est le suivant :

1. A partir d'un point  $x_0$ , on effectue un déplacement aléatoire
2. Si le déplacement mène à  $x$  tel que  $f(x) < f(x_0)$  alors  $x$  est accepté, sinon il est accepté ou rejeté selon une certaine probabilité.

Cette méthode définit une variable appelée température qui agit sur la probabilité d'acceptation de la nouvelle solution. Plus la température a une valeur élevée, plus il est possible d'admettre un point impliquant une grande perte d'efficacité par rapport à la fonction objectif. La température initiale de la simulation est généralement élevée de façon à favoriser la capacité d'exploration de l'espace des solutions. La capacité de déplacement des points diminue au fur et à mesure de l'optimisation qui se concentre de plus en plus sur des déplacements efficaces pour la minimisation de la fonction.

**Les algorithmes évolutionnaires** On distingue trois grandes classes d'algorithmes évolutionnaires : les algorithmes génétiques, les stratégies d'évolution et la programmation évolutive. Ces méthodes se distinguent par leur représentation de l'information et par leur stratégie d'évolution d'une génération à l'autre. Un algorithme évolutionnaire est composé de trois éléments fondamentaux :

- une population constituée d'individus représentant des solutions potentielles
- un mécanisme d'évaluation des individus permettant de mesurer l'adaptation d'un individu à son environnement

- un mécanisme d'évolution de la population permettant d'éliminer certains individus et d'en créer de nouveaux

Comme exemple d'algorithme évolutionnaire, nous allons présenter le fonctionnement général des algorithmes génétiques. On considère une population de points répartis dans l'espace de recherche. Le processus d'optimisation est constitué des étapes suivantes :

- Chaque point est un *individu* codé à la manière d'un gène.
- A chaque individu est associée une *efficacité* qui correspond à la valeur de l'individu par rapport au problème (dans notre cas, minimiser la fonction  $f$ ).
- Une *reproduction* des individus est effectuée proportionnellement à leur efficacité de façon à ce que les individus les plus intéressants soient plus représentés dans la population. Plus un individu est efficace, plus il est présent dans la population et donc plus il a de chances de transmettre ses gènes par l'un des deux mécanismes suivants.
- La phase de *croisement* permet de générer de nouveaux individus dans la population en mélangeant les gènes de 2 individus.
- La phase de *mutation* permet de générer un nouvel individu à partir d'un seul individu en modifiant une partie de ses gènes de manière aléatoire.
- En dernier lieu intervient la phase de *sélection* des individus. Deux approches sont possibles : la conservation de tous les individus (avec suppression des doublons) ou la suppression d'un certain nombre d'individus (par exemple les moins performants) pour restreindre la population.

Les phases de reproduction, croisement, mutation et sélection sont répétées jusqu'à atteindre un critère d'arrêt. L'optimum proposé sera alors l'individu le plus performant de la population.

### 2.2.3 Choix d'une méthode d'optimisation : compromis exploration-exploitation

La notion d'efficacité d'une méthode d'optimisation se rapporte généralement à deux objectifs : la vitesse et la précision. La capacité d'exploitation est l'aptitude d'une méthode à utiliser des résultats déjà obtenus pour faire converger l'algorithme rapidement. Cette vitesse est souvent mesurée en nombre d'évaluations de la fonction objectif, qui est la plupart du temps la partie la plus gourmande en temps de calcul. La capacité d'exploration est l'aptitude d'une méthode à explorer avec efficacité l'espace d'état de façon à trouver un résultat le plus précis possible. Cette précision est déterminée par la distance entre l'optimum trouvé par l'heuristique et l'optimum réel, soit du point de vue de la solution, soit de celui de la valeur. Ces deux objectifs sont bien souvent contradictoires. En effet une méthode donnant un résultat avec précision nécessite une recherche exploratoire très importante de façon à ne laisser aucune partie de l'espace des états non visitée. Dans le cas d'une méthode à forte capacité d'exploitation, le processus de convergence est accéléré au risque de voir la méthode trompée et dirigée dans une zone de l'espace non optimale. Le choix d'une méthode d'optimisation dépend de l'aspect que l'on cherche à privilégier.

Dans le cas où le calcul d'optimisation doit être limité en temps de calcul, le recuit simulé avec une température restant longtemps haute ne pourra pas répondre à nos attentes. Le problème d'optimisation qui nous concerne dans le cadre de cette thèse nécessite de limiter le temps passé à rechercher la solution nous avons donc privilégié le choix d'une méthode à convergence rapide (Nelder-Mead) à laquelle nous avons ajouté une phase d'initialisation adaptée à notre problème de façon à guider la recherche dans des zones paraissant les plus susceptibles de contenir le minimum global. Cette approche sera détaillée au chapitre 4.

## 2.3 Les problèmes d'optimisation multi-objectifs

### 2.3.1 Problématique

Un problème d'optimisation multi-objectifs (ou multi-critères) peut être décrit mathématiquement de la façon suivante :

$$\text{minimiser } \vec{f}(\vec{x}) = \begin{bmatrix} f_1(\vec{x}) \\ f_2(\vec{x}) \\ \dots \\ f_n(\vec{x}) \end{bmatrix} \quad \vec{x} \in R^N \quad (2.2)$$

avec

$$\begin{aligned} \vec{g}(\vec{x}) &\leq 0, \quad \vec{g}(\vec{x}) \in R^m \\ \vec{h}(\vec{x}) &= 0, \quad \vec{h}(\vec{x}) \in R^p \end{aligned}$$

Dans cette expression du problème,  $\vec{x}$  est le vecteur des variables de décision,  $N$  désigne le nombre de variables,  $n$  le nombre de fonctions objectifs,  $m$  et  $p$  le nombre de contraintes.

Les problèmes d'optimisation qui ont plus d'une fonction objectif sont courants dans tous les domaines. Dans ce type de problèmes, les objectifs à optimiser sont souvent en conflit les uns avec les autres, ce qui signifie qu'il n'existe pas de définition de la solution optimale. Le but de l'optimisation est alors de proposer un ensemble de solutions satisfaisantes pour lesquelles on ne pourra effectuer une opération de classement ou bien de trouver une solution parmi celles-ci qui soit un bon compromis au regard des différents objectifs.

### 2.3.2 Optimalité au sens de Pareto

La ou les solutions proposées pour un problème d'optimisation multi-critères ne sont pas optimales dans le sens où elles ne minimisent pas toutes les fonctions objectifs du problème. En effet, une telle solution n'existe pas pour la plupart des problèmes multi-critères. Une notion spécifique d'optimalité a donc été définie spécifiquement pour les problèmes d'optimisation multi-objectifs : la Pareto-optimalité<sup>1</sup>. Cette notion s'accompagne de la définition d'un certain nombre de termes permettant de comparer les solutions entre elles.

---

1. du nom de Vilfredo Pareto, sociologue et économiste italien qui a introduit le concept d'optimum de Pareto en économie qui correspond à un état dans lequel on ne peut améliorer le bien-être d'un individu sans détériorer celui d'un autre.

**Définition 3** Soit deux vecteurs  $\vec{x}, \vec{y} \in \mathbb{R}^N$ , on dit que  $\vec{x} \leq \vec{y}$  si  $\forall i \in \{1, \dots, n\} x_i \leq y_i$  et que  $\vec{x}$  domine  $\vec{y}$  (noté  $\vec{x} < \vec{y}$ ) si  $\vec{x} \leq \vec{y}$  et  $\vec{x} \neq \vec{y}$ .

**Définition 4** Un vecteur de décision  $\vec{x}^* \in S$  (ensemble des solutions réalisables)  $\subset \mathbb{R}^N$  est efficace ou **Pareto-optimal** s'il n'existe aucun  $\vec{x}' \in S$  tel que  $\vec{f}(\vec{x}') < \vec{f}(\vec{x}^*)$ .

**Définition 5** Le **front de Pareto**  $PF^*$  (également appelé **surface de compromis**) est défini par :

$$PF^* = \{ \vec{f}(\vec{x}) \in \mathbb{R}^N \mid \vec{x} \text{ est Pareto-optimal} \}$$

Un certain nombre de notions annexes ont été définies dans le domaine de l'optimisation multi-objectifs comme l'optimalité locale au sens de Pareto, l'optimalité lexicographique, l'optimalité extrême. Pour plus de détails sur le sujet nous invitons le lecteur à se référer à [CS02].

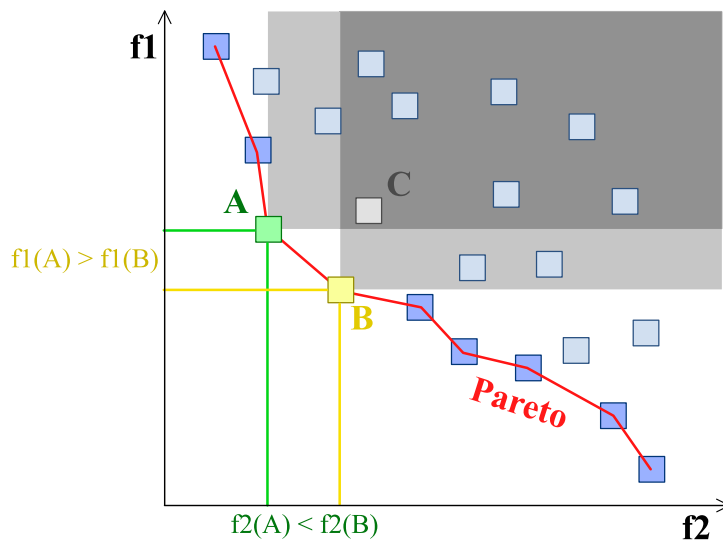


FIGURE 2.4: **Optimalité au sens de Pareto** : exemple pour un problème d'optimisation à 2 fonctions objectif : Les solutions A et B sont des solutions Pareto-optimales, puisqu'elles ne sont dominées par aucune autre solution. Elle font donc partie du front de Pareto. La solution C est dominée à la fois par A et B (zones grisées) puisque elles améliorent simultanément les 2 fonctions objectifs.

### 2.3.3 Méthodes de résolution multicritères

On rencontre deux types de classification des méthodes de résolution de problèmes multi-objectifs. Un premier classement adopte le point de vue utilisateur (décideur). Il distingue les trois approches suivantes :

- Les méthodes *a priori* pour lesquelles l'utilisateur définit le compromis qu'il désire réaliser avant de lancer la méthode de résolution. Elles transforment le problème multi-critère en un problème monocritère, en pondérant l'ensemble des critères initiaux.

- Les méthodes *progressives*, pour lesquelles l'utilisateur affine son choix des compromis au fur et à mesure du déroulement de l'optimisation. Ces méthodes ont l'inconvénient de monopoliser l'attention du décideur durant le processus d'optimisation.
- Les méthodes *a posteriori* pour lesquelles il n'est plus nécessaire, pour le concepteur, de modéliser ses préférences avant l'optimisation. Ces méthodes produisent un ensemble de solutions directement transmis au décideur qui choisira alors *a posteriori* une solution de compromis parmi celles proposées.

Le deuxième classement se place du point de vue du développeur et se base sur les notions d'agrégation et d'optimum de Pareto :

- Les méthodes agrégées transforment un problème multi-objectifs en un problème mono-objectif.
- Les méthodes non agrégées qui possèdent un processus de recherche qui traite séparément les objectifs.
- Les méthodes fondées sur Pareto qui s'appuient sur la notion de dominance au sens de Pareto et qui privilégie une recherche satisfaisant au mieux tous les objectifs.

Les méthodes développées dans le domaine sont nombreuses et les possibilités d'hybridation des techniques importantes [Ber01, CS02]. Nous n'aborderons donc ici les différentes approches que de façon très générale, et évoquons quelques méthodes pour les illustrer. En particulier nous développerons plus en détail les approches agrégatives que nous avons utilisées pour notre problème.

### 2.3.3.1 Les méthodes agrégées

L'ensemble de ces méthodes repose sur l'affirmation suivante : tout décideur essaye inconsciemment de maximiser une fonction d'utilité  $U$ .

$$U = U(f_1, f_2, \dots, f_n)$$

Nous allons présenter les techniques les plus courantes d'agrégation des objectifs.

**La méthode de pondération** Une approche classique pour convertir un problème d'optimisation multi-critères en un problème d'optimisation scalaire est de minimiser une combinaison linéaire des différentes fonctions objectif. En d'autres termes  $n$  poids  $w_i$  sont choisis tel que  $w_i \geq 0$  et  $\sum_{i=1}^n w_i = 1$ . La résolution du problème multi-critères se ramène alors à la résolution du problème suivant :

$$\text{Minimiser } \sum_{i=1}^n w_i f_i(\vec{x}), \quad \vec{x} \in S \subset R^N$$

Il résulte immédiatement que le minimum global  $\vec{x}^*$  de ce problème est un optimum de Pareto pour le problème. Si ce n'était pas le cas, il existerait un  $\vec{x}$  qui améliore au moins une des fonctions objectif sans pénaliser les autres et qui donc produirait une valeur inférieure pour la somme pondérée des  $f_i$ .

Cette méthode est simple à mettre en oeuvre et d'une grande efficacité. Elle présente l'avantage de pouvoir utiliser directement les algorithmes classiques dédiés au problème



d'optimisation mono-objectif. La principale difficulté de cette approche est la détermination des poids correspondant à chaque critère. Cette difficulté peut être surmontée aisément en laissant la possibilité de faire varier les poids de façon à constater l'influence de tel ou tel objectif sur le résultat.

Cependant, cette approche présente plusieurs limites. L'interaction entre les différents critères ne peut pas être représentée. L'utilisation de la somme pondérée se heurte également à la sensibilité au changement d'échelle et à la compensation entre critères. D'autre part, en faisant varier les poids de chaque contrainte, on peut obtenir un échantillonnage de solutions Pareto-optimale permettant d'obtenir une approximation du front de Pareto du problème. Cependant cette méthode ne permet pas d'atteindre toutes les solutions de la surface de compromis lorsque celle-ci n'est pas convexe. Cette limitation est illustrée sur la figure 2.5. Dans les cas d'une optimisation à 2 objectifs, la méthode de pondération consiste à minimiser la constante  $C$  de l'équation de droite suivante :  $f_2(\vec{x}) = -\frac{w_1}{w_2} \cdot f_1(\vec{x}) + C$ . La méthode de pondération consiste donc à trouver le coefficient  $C$  pour lequel cette droite est tangente avec l'espace  $S$  des solutions réalisables. La solution optimale est alors représentée par le point de tangence. La modification des poids permet alors d'atteindre différentes solutions de la frontière de Pareto. En revanche, il est impossible de trouver une droite tangente en une solution de la frontière de Pareto qui n'appartient pas à l'enveloppe convexe de l'ensemble des solutions. Dans le cas illustré par la figure, les solutions  $M_1$  et  $M_2$  sont atteignables en choisissant les poids correspondant respectivement aux droites  $d_1$  et  $d_2$ , alors qu'il n'existe pas de combinaison de poids permettant d'atteindre la solution  $M_3$  par cette méthode. Enfin la méthode de pondérations, comme les autres méthodes agrégatives, implique que tout sous-ensemble d'objectifs soit préférentiellement indépendant dans l'ensemble  $F$  des objectifs. Par exemple supposons que nous ayons affaire à un problème d'optimisation portant sur 3 critères. Si le premier critère est correctement satisfait alors on favorisera une solution optimisant le deuxième critère. Dans le cas contraire on favorisera une solution optimisant le troisième critère. Dans ce problème, il existe des dépendances préférentielles entre les critères. Ce type de dépendances ne peut être traduit dans l'expression d'une seule fonction objectif à optimiser.

**La méthode  $\epsilon$ -contrainte** Cette méthode est basée sur la minimisation d'un objectif  $f_i$  en considérant que les autres objectifs  $f_j$  avec  $j \neq i$  doivent être inférieurs à une valeur  $\epsilon_j$ . En général, l'objectif choisi  $f_i$  est celui que le décideur souhaite optimiser en priorité.

$$\text{Minimiser } f_i(\vec{x}) \text{ avec } \forall j \neq i \ f_j(\vec{x}) \leq \epsilon_j \text{ et } \vec{x} \in S \subset R^N$$

Par rapport à la méthode par pondération, cette approche a l'avantage de ne pas être trompée par les problèmes non convexes (voir figure 2.6). En contrepartie, pour obtenir un bon échantillonnage de la surface de compromis, il est nécessaire de lancer le processus de résolution un grand nombre de fois.

**Programmation par but** Le décideur fixe un but  $T_i$  à atteindre pour chaque objectif  $f_i$ . Ces valeurs sont ensuite ajoutées au problème comme des contraintes supplémentaires.



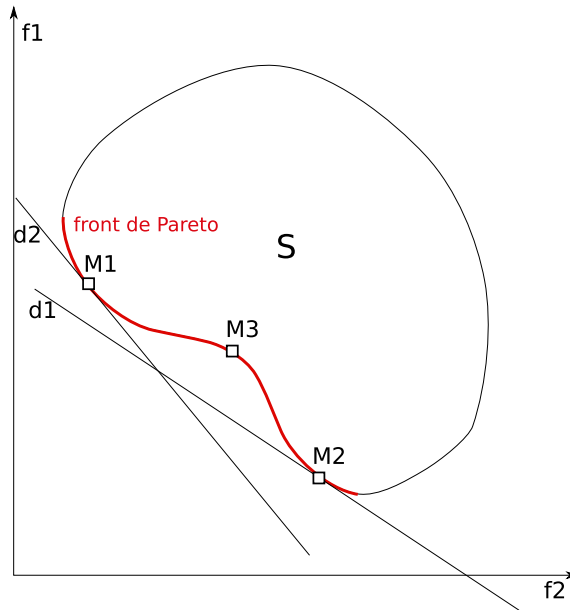


FIGURE 2.5: **Méthode de pondération** :  $S$  représente l'ensemble des valeurs atteintes pour  $(f_1, f_2)$  dans le domaine de recherche. Les point  $M_1$  et  $M_2$  sont des solutions Pareto-optimales atteignables par la méthode de pondération avec différents poids sur les fonctions objectif. Par contre, le point  $M_3$  du front de Pareto ne peut pas être détecté par cette méthode.

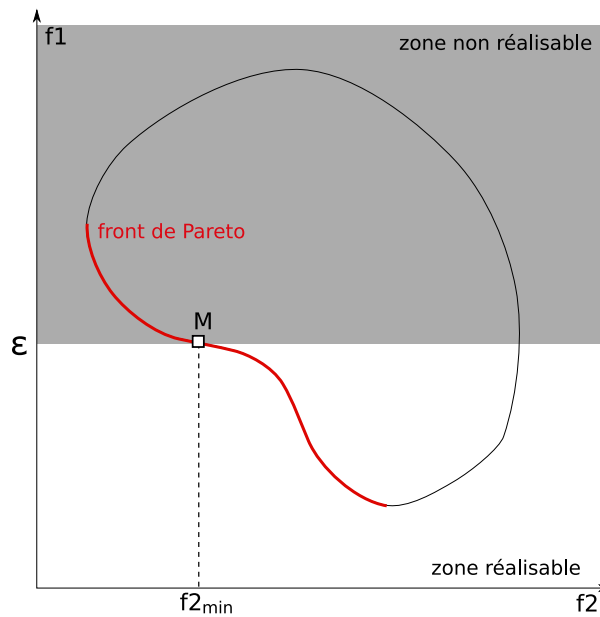


FIGURE 2.6: **Méthode  $\epsilon$ -contrainte** : La pose de la contrainte pour la fonction objectif  $f_1$  transforme le problème en problème d'optimisation simple avec pour objectif la fonction  $f_2$ . Le point  $M$ , appartenant à la zone non convexe du front de Pareto du problème multi-critère est alors atteint par le processus d'optimisation.

La nouvelle fonction objectif est modifiée de façon à minimiser la somme des écarts entre

les résultats et les buts à atteindre :

$$\text{Minimiser } \sum_{i=1}^k |f_i(\vec{x}) - T_i|, \quad \vec{x} \in S \subset R^N$$

Nous pouvons reprendre les critiques faites pour la somme pondérée. La méthode est facile à mettre en œuvre, mais la définition du point de référence représentant les objectifs à atteindre joue un rôle fondamental. Si celui-ci est mal choisi la recherche peut s'avérer laborieuse. Cette méthode a l'avantage de fournir un résultat même si un mauvais choix initial a conduit le décideur à donner un ou plusieurs buts non réalisables.

### 2.3.3.2 Méthodes traitant séparément les objectifs

Les méthodes agrégatives règlent les problèmes d'optimisation multi-objectifs en les transformant en un problème d'optimisation classique mono-critère, quitte à dénaturer leur expression initiale. Il existe d'autres méthodes traitant séparément les fonctions objectifs, telles qu'elles sont exprimées à l'origine. Les méthodes que nous évoquons ici n'exploitent pas la notion de dominance au sens de Pareto.

**La méthode lexicographique :** La méthode lexicographique consiste à ordonner les différents critères selon leur importance dans la décision finale. Le processus d'optimisation est lancé en prenant comme fonction objectif la fonction correspondant au critère classé en premier. La valeur optimale obtenue à l'issue de ce processus d'optimisation est alors utilisée pour la pose d'une nouvelle contrainte : seront prises en considérations seules les solutions correspondant à la valeur optimale pour ce premier critère. Ce processus est ensuite répété pour chacune des autres contraintes dans l'ordre de leur classement. Le principal inconvénient de cette méthode est l'importance prépondérante accordée aux critères classés en premier.

**L'algorithme VEGA :** L'algorithme VEGA (Vector Evaluated Genetic Algorithm) est une extension d'un algorithme génétique pour la résolution de problèmes multi-objectifs. La différence avec un algorithme génétique simple s'effectue lors de l'étape de sélection. Elle consiste à sélectionner, pour chaque critère un nombre fixe d'individus les plus performants pour le critère considéré. Les sous-populations sont alors mélangées afin d'appliquer les opérateurs de croisement et de mutation pour la génération suivante. Cette méthode a tendance à proposer des individus spécialisés dans un des objectifs, alors que les individus ayant une performance générale acceptable mais aucun critère fort sont éliminés.

### 2.3.3.3 Les méthodes Pareto

Les approches Pareto exploitent les notions de dominance et de front de Pareto de façon à proposer une population de solutions représentant au mieux le front de Pareto du problème (couverture complète et répartition uniforme des solutions sur le front). L'approche évolutionnaire présentée dans la section 2.2.2.2 dans le contexte mono-objectif permet la

mise en oeuvre d'approches non Pareto (agrégation des objectifs) mais également l'implémentation d'approches Pareto.

En effet certains mécanismes de sélection implémentent la relation de dominance. La technique du ranking par exemple utilise la relation de dominance de Pareto pour affecter un rang aux individus. Tous les individus non dominés de la population reçoivent le rang 1. Ces individus sont alors temporairement retirés de la population pour déterminer les individus de rang 2, non dominés dans la population restante. La valeur d'adaptation de chaque individu correspond à son rang dans la population. L'évaluation d'un individu ne dépend alors pas uniquement de lui-même mais aussi du reste de la population. Comme nous l'avons évoqué dans la section 2.2.3 une méthode d'optimisation mono-critère est caractérisée à la fois par sa capacité à exploiter les calculs nécessaires pour l'optimisation afin de converger rapidement vers une solution, et sa capacité à explorer l'espace des solutions de façon à éviter les minima locaux. Dans le cas multi-critère, lorsqu'on cherche à déterminer le front de Pareto, la capacité d'exploration d'une méthode permet non seulement d'éviter les minima locaux mais aussi de découvrir l'intégralité du front de Pareto. Plus de détails sur les approches Pareto dans les algorithmes évolutionnaires peuvent être trouvés dans [CS02] et [Bar03].

### 2.3.4 Discussion

L'approche agrégative pour la résolution des problèmes d'optimisation multi-critères est certainement la plus courante et la plus intuitive. Que ce soit sous forme pondérée ou sous forme de distance à un but, cette transformation permet d'utiliser facilement les méthodes d'optimisation issues de l'optimisation mono-objectif. L'optimisation multi-objectif peut conduire à approximer le front de Pareto. Les méthodes par agrégation obligent alors à relancer le processus d'optimisation à de nombreuses reprises avec différentes valeurs des paramètres. En revanche, lorsque l'objectif n'est pas de proposer un échantillonnage des compromis Pareto-optimaux mais d'en sélectionner un parmi ceux-là, la méthode agrégative est une bonne solution à condition que l'on arrive à exprimer le rôle des différents critères à prendre en compte dans une seule fonction objectif. Elle nécessite donc une bonne connaissance du problème a priori. L'approche agrégative est celle que nous avons sélectionnée pour notre problème d'optimisation de placement d'outil chirurgical, puisque ce qui intéresse le praticien n'est pas de connaître toutes les stratégies correspondant à un compromis optimal selon l'importance que l'on accorde au différents critères, mais de trouver *la* stratégie correspondant aux différents critères en tenant compte du rôle et de l'importance qu'il leur accorde. Nous aurions pu nous limiter à la méthode de pondération, en laissant la possibilité à l'utilisateur de jouer sur les poids des différentes contraintes. Nous avons préféré cependant ne pas restreindre les possibilités en laissant librement l'utilisateur définir la fonction objectif agrégative en fonction des différents critères. Selon la méthode choisie, il est alors possible d'accéder à des solutions appartenant à une partie non convexe de l'espace des solutions. Les possibilités offertes pour la formalisation du problème d'optimisation multi-objectifs et en particulier de la fonction objectif agrégative des différents critères seront détaillées au chapitre suivant.



## Chapitre 3

# Formalisation des connaissances

### Sommaire

---

<b>3.1</b>	<b>Construction d'un terme représentant une contrainte . . . .</b>	<b>54</b>
3.1.1	Types manipulés . . . . .	54
3.1.2	Opérateurs . . . . .	55
3.1.3	Représentation des termes sous forme d'arbres . . . . .	56
3.1.4	Exemples de termes . . . . .	57
<b>3.2</b>	<b>Connaissances à formaliser . . . . .</b>	<b>58</b>
3.2.1	Distinction contraintes strictes / contraintes souples . . . . .	58
3.2.2	Choix de la nature d'une contrainte à formaliser . . . . .	59
3.2.3	L'introduction de données complémentaires . . . . .	61
<b>3.3</b>	<b>Définition de contraintes grâce au formalisme XML . . . . .</b>	<b>62</b>
3.3.1	Spécificité des données complémentaires . . . . .	63
3.3.2	Reformulation du problème défini par les contraintes strictes . .	64
3.3.2.1	Définition de termes correspondant à des contraintes strictes . . . . .	66
3.3.3	Contraintes souples et optimisation . . . . .	67
3.3.4	Conformité du fichier de contrainte . . . . .	68
3.3.4.1	Désactivation d'une contrainte dont le terme n'est pas reconnu . . . . .	68
3.3.4.2	Fichier de contraintes minimal . . . . .	69
<b>3.4</b>	<b>Fichier de contraintes correspondant à la radiofréquence hépatique . . . . .</b>	<b>70</b>
3.4.1	Les contraintes à formaliser . . . . .	70
3.4.2	Définition des contraintes strictes . . . . .	73
3.4.2.1	"L'aiguille traverse la peau et sa pointe doit être placée dans la tumeur traitée" . . . . .	73
3.4.2.2	"Le volume traité doit englober la totalité du volume de la tumeur accompagnée d'une marge de 5 mm." . . .	73
3.4.2.3	"L'aiguille ne doit pas endommager les organes vitaux du patient" . . . . .	73

---

3.4.2.4	"La longueur de l'aiguille doit permettre d'atteindre la tumeur " . . . . .	75
3.4.2.5	"La trajectoire de l'aiguille ne doit pas être trop tangente à la surface du foie" . . . . .	75
3.4.2.6	"La trajectoire de l'aiguille doit inclure une portion de foie sain" . . . . .	76
3.4.3	Les contraintes strictes : choix de formalisation . . . . .	76
3.4.4	Définition des contraintes souples . . . . .	77
3.4.4.1	"Le volume de tissus sains nécrosés par le traitement doit être minimal" . . . . .	77
3.4.4.2	"La trajectoire de l'aiguille doit être la plus directe possible" . . . . .	77
3.4.4.3	"Les trajectoires frôlant les organes vitaux doivent être évitées" . . . . .	78
3.4.5	L'agrégation des contraintes souples . . . . .	79
3.4.5.1	Mise à l'échelle des différentes contraintes . . . . .	79
3.4.5.2	Contrainte représentant la fonction objectif . . . . .	82
<b>3.5</b>	<b>La formalisation des contraintes : conclusions et perspectives</b>	<b>83</b>

---

DANS ce chapitre, nous allons décrire le langage développé permettant de décrire formellement les contraintes intervenant dans la décision d'une stratégie opératoire pour une opération d'ablation percutanée. L'ablation par radiofréquence est pratiquée localement au CHU de Strasbourg par le Professeur Gangi et le Docteur Buy qui sont des radiologues interventionnels expérimentés et qui ont été pour nous une source d'information indispensable. D'autre part, la radiofréquence est une des techniques d'ablation mini-invasives les plus populaires et est particulièrement documentée dans le cas des tumeurs hépatiques depuis quelques années déjà. Nous avons donc commencé ces travaux de thèse en nous concentrant sur la RFA des tumeurs dans le foie. Nous avons ainsi regroupé les informations disponibles dans la littérature médicale et issues de différents entretiens avec des praticiens afin d'extraire les règles de la technique du point de vue de la planification. Le Pr Gangi et le Dr Buy nous ont accordé plusieurs entretiens au cours desquels nous avons pu les interroger sur leurs pratiques et sur les raisons qui motivent leur choix stratégique spécifique à chaque intervention. Nous nous sommes également entretenus avec d'autres spécialistes (Pr Mutter, chirurgien, Strasbourg et Pr Pereira, radiologue interventionnel, Tuebingen) qui nous ont permis d'aborder les points de vue de différents spécialistes. En complément de ces entretiens, nous avons assisté à une dizaine d'opérations de radiofréquence pratiquées par les radiologues du CHU de Strasbourg, et notamment aux phases de planification précédant l'opération, au cours desquelles ils ont pris le temps de nous expliquer leurs décisions spécifiques à chaque opération. Au fil de ces différents entretiens et opérations qui se sont étalés sur plusieurs mois, nous nous sommes rendus compte que les techniques peuvent varier considérablement d'un centre médical à l'autre, d'un praticien à l'autre et selon le matériel disponible localement (aiguilles et générateurs de radiofréquence, modalités d'imagerie). Depuis le début de nos travaux, les radiologues du CHU de Strasbourg ont également changé de technique de prédilection, puisque dernièrement ceux-ci ont principalement pratiqué des interventions de cryoablation. Tout cela a motivé l'orientation de nos travaux : les contraintes régissant les décisions stratégiques ne doivent pas être figées. Celles-ci doivent être facilement modifiables et adaptables à la technique et au praticien concerné. Les règles de planification, clairement énoncées dans le domaine ou issues de l'expérience du spécialiste, sont alors considérées comme des données indépendantes du programme qui les interprète afin de fournir une solution.

Le premier axe de ce travail de thèse est la définition d'un langage permettant d'exprimer les contraintes de toutes sortes qui dirigent la planification de l'intervention envisagée. Contrairement à la conception en CAO, pour la planification d'opération chirurgicale les contraintes sont relativement similaires d'une intervention à l'autre et sont peu modifiées au cours de la planification. Nous avons donc choisi de représenter les contraintes sous forme d'un script interprété au début de la planification, tout en laissant la possibilité à l'utilisateur d'intervenir sur certains paramètres intégrés dans les contraintes en cours de planification pour ajuster le choix définitif. Les critères de choix d'une stratégie d'insertion énoncés par les spécialistes en langage naturel sont traduisibles par une combinaison d'opérations géométriques. Dans un premier temps nous détaillerons comment il est possible de construire un terme représentant une opération géométrique sur différents objets faisant

partie des données de base. Puis nous détaillerons comment ces termes sont utilisés pour définir des contraintes qui formalisent la problématique de planification d'une stratégie d'insertion d'outil chirurgical. Nous présenterons alors le langage XML utilisé pour mettre en forme les contraintes et les termes correspondant. Enfin nous présenterons les choix de formalisation dans le cas de la planification d'une opération de radiofréquence percutanée sur une tumeur hépatique.

## 3.1 Construction d'un terme représentant une contrainte

La définition des contraintes géométriques dans le cadre d'une intervention chirurgicale nécessite de pouvoir manipuler des objets de différents *types* ou *sortes* grâce à des *fonctions* ou *opérateurs*. C'est en combinant ces différents objets grâce aux opérateurs que l'on peut construire des *termes* permettant de définir une contrainte. Nous allons maintenant détailler la nature des objets manipulés ainsi que les opérations associées puis nous montrerons des exemples de ce que nous pouvons exprimer grâce à ces différents éléments.

### 3.1.1 Types manipulés

Les objets pouvant être désignés lors de la construction d'une contrainte peuvent être de différentes natures. Il est possible de manipuler des objets de type simple (**booléens** et **réels**<sup>1</sup>), mais également de types composés. Nous énumérons ici ces types et nous illustrons chacun d'entre eux par des exemples :

- Les **points** (*Point*) définissant une position dans l'espace par ses coordonnées (ex : le centre de la tumeur, la pointe de l'outil courant).
- Les **demi-droites** (*Ray*) qui désignent principalement une trajectoire potentielle pour un outil (ex : la trajectoire courante de l'outil).
- Les **formes** (*Shape*) qui désignent les objets tridimensionnels (ex : la peau, la tumeur, la zone de nécrose de l'aiguille courante). Ce type englobe à la fois les surfaces et les volumes.
- Les **stratégies** (*Strategy*) qui définissent un choix pour les 6 paramètres correspondant à l'inconnue du problème. Cette stratégie est constituée d'une demi-droite (3 paramètres pour le point d'origine de la demi-droite, 2 pour les coordonnées polaires du vecteur directeur) représentant le placement de l'outil ainsi qu'un réel représentant l'intensité du traitement et permettant de déduire l'ampleur de la zone d'effet (1 paramètre).

Notons que les types *Point*, *Ray* et *Shape* sont regroupés dans un type générique représentant les **objets géométriques** (*GeomObject*), cela correspond au niveau de l'implantation à une relation d'héritage.

Les données de base manipulables par la définition de contraintes sont des objets de type *Forme* qui représentent l'anatomie du patient. Ces objets sont manipulables dans la définition d'une contrainte en utilisant directement leur nom. A l'heure actuelle, les données

---

1. les entiers ne sont pas considérés car utilisables en tant que aux réels



définies en tant que structure anatomiques sont les suivantes : *skin, liver, kidney, lung, heart, spleen, pancreas, intestine, vessels* et *tumor*. Ces données peuvent être accessibles à la fois sous forme volumique et sous forme surfacique. En effet, on dispose de données sous forme d'ensembles de voxels issus de la segmentation des images scanner et pouvant représenter le volume de l'objet, mais également de maillages représentant leur surface (construits à partir des données voxel). Cependant du point de vue de la définition des contraintes, cette distinction n'a pas lieu d'être puisque celle-ci doit être indépendante du type de données manipulées qui reste interne au solveur. Nous avons donc regroupé les deux types de données sous le même type *Forme* qui constitue une barrière d'abstraction. Pour chaque organe du patient, on dispose alors des 2 types de données qui sont utilisées par le solveur en version surfacique ou volumique selon le cas.

### 3.1.2 Opérateurs

TABLE 3.1: Liste des opérateurs définis et prototypes correspondant

Opérateurs booléens	or, and not	$bool * bool \rightarrow bool$ $bool \rightarrow bool$
Opérateurs numériques	plus, mult minus, inverse pow lower min, max ifthenelse	$float * float \rightarrow float$ $float \rightarrow float$ $float * float \rightarrow float$ $float * float \rightarrow bool$ $float * float \rightarrow float$ $bool * float * float \rightarrow float$
Opérateurs géométriques	distmin center angle intersection visible enlarge, reduce boundingball union	$GeomObject * GeomObject \rightarrow float$ $Shape \rightarrow Point$ $Ray * Shape \rightarrow float$ $Ray * Shape \rightarrow Point$ $Shape * Shape \rightarrow Shape$ $Shape * float \rightarrow Shape$ $Shape \rightarrow Shape$ $Shape * Shape \rightarrow Shape$
Opérateurs spécifiques à la stratégie	toolTip toolTrajectory toolEffect toolInsertionPoint cross target cover	$Strategy \rightarrow Point$ $Strategy \rightarrow Ray$ $Strategy \rightarrow Shape$ $Strategy \rightarrow Point$ $Strategy * Shape \rightarrow Bool$ $Strategy * Shape \rightarrow Bool$ $Strategy * Shape \rightarrow Bool$

Un certain nombre d'opérateurs ont été définis afin de pouvoir créer un maximum de contraintes potentielles. Ceux-ci ne sont pas exhaustifs et il est probable que la définition

d'autres opérateurs pourrait être nécessaire pour créer de nouvelles contraintes dans le futur, mais ceux que nous présentons ici permettent déjà de définir beaucoup de contraintes. Les opérateurs existants sont énumérés dans la tableau 3.1.

Un minimum d'opérateurs classiques sur les flottants et les booléens sont définis. Les opérateurs *or*, *and*, *plus*, *mult* sont des opérateurs binaires, cependant ils sont aussi utilisables avec un nombre quelconque de paramètres (booléens ou flottants selon le cas) de façon à rendre l'écriture des contraintes plus lisibles.

Les opérateurs géométriques permettent d'effectuer les calculs les plus courants à partir des différents objets géométriques constituant la scène 3D. L'opérateur *union* permet également de prendre un nombre quelconque de paramètres au lieu d'être effectué en cascade sur ses opérands pour simplifier l'écriture.

La dernière catégorie d'opérateurs permet d'accéder à certaines informations spécifiques à la stratégie recherchée de façon à la manipuler facilement. Nous noterons que tous ces opérateurs prennent comme paramètre un objet de type *Strategy*. Pour simplifier l'écriture et puisque dans le cadre de l'écriture de contraintes, l'objet de type *Strategy* désigne systématiquement la stratégie d'insertion étudiée pour l'outil courant, ce paramètre est omis dans la description des contraintes. Ainsi on pourra utiliser directement *toolInsertionPoint* plutôt que *toolInsertionPoint(currentNeedle)*.

### 3.1.3 Représentation des termes sous forme d'arbres

Les termes définissent des opérations géométriques constitués d'opérateurs et d'opérands. La représentation d'opérations sous forme d'arbres est classique, nous l'utiliserons donc dans la suite de ce mémoire pour illustrer les contraintes et leur fonctionnement. Cette représentation est associée à l'utilisation d'un vocabulaire que nous allons détailler.

Les *noeuds internes* de l'arbre représentent des opérateurs, les *noeuds feuilles* des données. Pour distinguer graphiquement les types de noeuds, les noeud feuilles sont doublement cerclés.

Le *type* d'un noeud est le type de la donnée dans le cas d'un noeud feuille, et le type de renvoi de l'opération dans le cas d'un noeud interne.

La *valeur* d'un noeud correspond à la valeur de la donnée dans le cas d'un noeud feuille. La valeur d'un noeud interne correspond à la valeur résultante de l'opération correspondant au sous-arbre dont ce noeud est la racine. La valeur d'un terme correspond à la valeur de la racine de l'arbre.

La valeur d'un terme dépend de la valeur des données. Elle est donc fixe tant que les données manipulées le sont et ne devra être évaluée qu'une seule fois<sup>2</sup>. Il est possible de manipuler des termes à valeur variable. Ces termes font intervenir l'inconnue du problème dans leur définition, c'est-à-dire qu'ils utilisent au moins un opérateur prenant en paramètre la stratégie recherchée. Ces *noeuds inconnus* se distinguent visuellement des autres grâce à un cerclage en gras.

---

2. nous verrons par la suite que les données peuvent dans certain cas changer de valeur en cours de planification

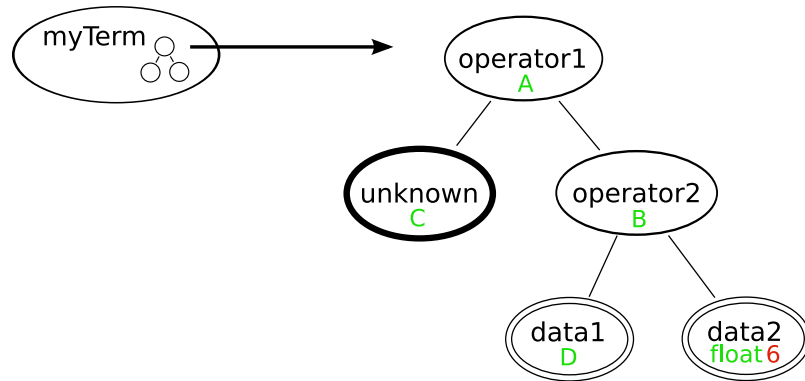


FIGURE 3.1: Représentation graphique des termes sous forme d'arbre

On considère le terme suivant, illustré par la figure 3.1 :

$$\text{operator1}(\text{unknown}, \text{operator2}(\text{data1}, \text{data2})) \quad (3.1)$$

Les types des noeuds peuvent être indiqués (en vert sur la figure), par exemple *data1* est de type *D*. Pour que la contrainte soit cohérente il est nécessaire que le prototype de l'opérateur d'un noeud interne corresponde au type de ses noeuds fils. Par exemple on peut déduire de la figure que l'*operator1* a pour prototype  $C * B \rightarrow A$ . La valeur d'un noeud peut également être précisée, par exemple *data2* est un réel qui vaut 6. Enfin le type du terme représenté sur la figure est le type de la racine, c'est-à-dire *A*. Un terme déjà défini peut être utilisé dans la définition d'un autre terme, il sera alors représenté par un noeud composite (*myTerm* sur la figure).

### 3.1.4 Exemples de termes

Nous allons maintenant montrer des exemples de termes représentant des opérations simples. Admettons que nous ayons besoin de manipuler la distance entre la tumeur et la surface du foie. Le terme correspondant est le suivant :

$$\text{distmin}(\text{liver}, \text{tumor}) \quad (3.2)$$

Ce terme est représenté sous forme d'un arbre sur la figure 3.2, l'opérateur *distmin* étant un noeud interne et les opérandes *tumor* et *liver* des feuilles.

Dans la section 3.1.2, nous avons évoqué à part les opérateurs spécifiques à la stratégie permettant de manipuler les inconnues de notre problème de façon intuitive. Nous avons vu que la stratégie en paramètre est omise dans le terme. C'est également le cas pour la représentation graphique sous forme d'arbre : les opérateurs *toolTip*, *toolTrajectory*, *toolEffect* et *toolInsertionPoint* sont représentés par des feuilles et les opérateurs *cross*, *target* et *cover* sont des noeuds internes à un seul noeud fils. Considérons le terme suivant représentant la distance entre la trajectoire de l'aiguille et le poumon droit :

$$\text{distmin}(\text{toolTrajectory}, \text{lung\_right}) \quad (3.3)$$

Pour une intervention donnée (un patient et une tumeur cible), la valeur du terme 3.2 est fixe. En revanche le terme 3.3 a une valeur variable en fonction de la stratégie courante étudiée à cause de l'utilisation de l'opérateur *toolTrajectory*.

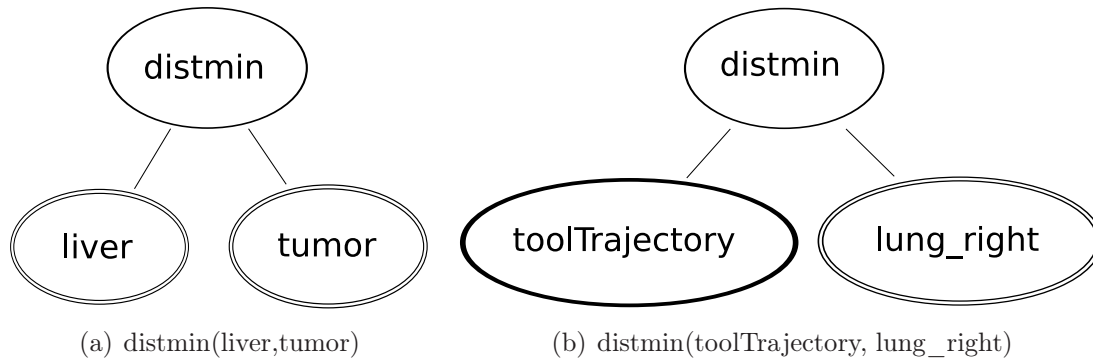


FIGURE 3.2: Exemples de terme

## 3.2 Connaissances à formaliser

Pour chaque intervention le praticien planifie une stratégie qui doit tenir compte de plusieurs objectifs et tenter de les concilier entre eux pour parvenir à une stratégie acceptable. Ces objectifs forment un problème d'optimisation multi-critères. La stratégie choisie pour l'opération correspond au meilleur compromis trouvé par rapport au problème. Afin de pouvoir fournir au praticien une proposition de stratégie optimale selon ses critères, il est nécessaire de fournir des outils pour formaliser le problème d'optimisation que l'on cherche à résoudre. En mathématiques les problèmes d'optimisation sont généralement définis par une ou plusieurs fonctions à optimiser ainsi qu'un certain nombre de contraintes d'égalité ou d'inégalité (cf chapitre précédent, section 2.3.1). Dans notre cas, nous avons distingué ces deux parties du problème sous les noms de *contraintes strictes* et *contraintes souples*.

### 3.2.1 Distinction contraintes strictes / contraintes souples

Considérons les contraintes suivantes exprimées en langage naturel :

- "L'outil ne *doit* pas blesser les organes vitaux"
- "L'outil ne *peut* pas traverser les os"
- "Le praticien *préfère* une trajectoire directe"
- "On *privilégie* un placement permettant de préserver au maximum les tissus sains"

Les deux premières contraintes expriment une impossibilité ou une interdiction. Ces contraintes sont de nature booléenne, elle doivent être respectées. Un placement d'outil qui respecte toutes ces contraintes est une solution du problème. Ces contraintes, que nous appellerons par la suite *contraintes strictes*, déterminent l'espace des solutions. A priori,

on s'attend à être confronté à un problème sous-contraint ; c'est-à-dire que l'espace des solutions résultant de l'application des contraintes strictes contient plusieurs solutions, et même plus vraisemblablement une infinité puisqu'on travaille dans un domaine continu. Il est alors nécessaire de pouvoir faire un classement parmi les différentes solutions afin de choisir la solution la plus adéquate à proposer au praticien. Les deux autres contraintes présentées expriment la notion de préférence. Ces *contraintes souples* sont les contraintes à satisfaire au mieux. Elles sont de nature numérique et représentent les fonctions objectif de notre problème d'optimisation multi-critères. La résolution du problème de planification défini par l'ensemble des contraintes strictes et souples revient à proposer une ou plusieurs stratégies qui soient à la fois *solutions* et *optimales*. Nous définissons ces deux termes de la façon suivante :

**Définition 6** Soit  $S \in R^6$  une stratégie d'insertion d'outil et  $C = \{C_{strict}, C_{soft}\}$  un ensemble de contraintes.  $S$  est une stratégie solution si  $\forall strict_i \in C_{strict}, strict_i(S)$  est vérifiée.

**Définition 7** Soit  $S \in R^6$  une stratégie d'insertion d'outil et  $C = \{C_{strict}, C_{soft}\}$  un ensemble de contraintes.  $S$  est une stratégie optimale si  $S$  est une solution et si  $\nexists S'$  tel que  $\forall soft_i \in C_{soft}, soft_i(S) < soft_i(S')$  ( $S$  est un optimum de Pareto par rapport à  $C_{soft}$ , l'ensemble des fonctions objectif).

### 3.2.2 Choix de la nature d'une contrainte à formaliser

Il n'est pas toujours évident de déterminer, à partir de la formulation naturelle d'une contrainte s'il s'agit de formaliser une contrainte stricte ou une contrainte souple. Nous avons vu que leur nature est différente : les contraintes strictes sont booléennes alors que les contraintes souples sont numériques. Cependant cette distinction grâce à leur nature n'est pas forcément justifiée puisque l'on passe facilement de l'une à l'autre. En effet une contrainte souple peut aisément être transformée en une contrainte stricte si l'on détermine une valeur à partir de laquelle la contrainte est éliminatoire. Le résultat de la contrainte est alors interprété comme la valeur booléenne *true* ou *false* selon qu'il est inférieur ou non à la valeur limite. Il en est de même pour les contraintes strictes. La nature booléenne d'une contrainte ne signifie pas qu'elle doit être respectée ; on peut imaginer une pénalité pour le processus d'optimisation dans le cas où la contrainte n'est pas respectée. Elle interviendrait alors en tant que contrainte souple. La figure 3.3 décrit le passage de l'un à l'autre sous forme de termes tels que nous les avons décrits dans la section précédente. Le passage d'un type de contrainte à l'autre n'est pas anodin. Alors qu'une contrainte décrite sous forme de contrainte stricte élimine des possibilités, la même contrainte construite en tant que contrainte souple permet d'appliquer une pénalité non disqualifiante aux possibilités en infraction.

Remarquons également qu'une contrainte exprimée verbalement peut se traduire par deux contraintes : l'une stricte et l'autre souple. En effet on peut imaginer qu'une contrainte souple à minimiser puisse prendre une valeur tellement importante qu'elle serait éliminatoire pour les stratégies qui l'atteignent. Prenons l'exemple de la contrainte suivante : "la

trajectoire de l'outil ne doit pas approcher les organes vitaux de trop près". Elle pourrait être traduite par la conjonction de deux contraintes : une contrainte stricte qui élimine de l'espace des solutions toutes les trajectoires approchant un organe vital à moins de 2mm et une contrainte souple qui donne une pénalité décroissante aux trajectoires approchant les organes vitaux de 2 à 5mm.

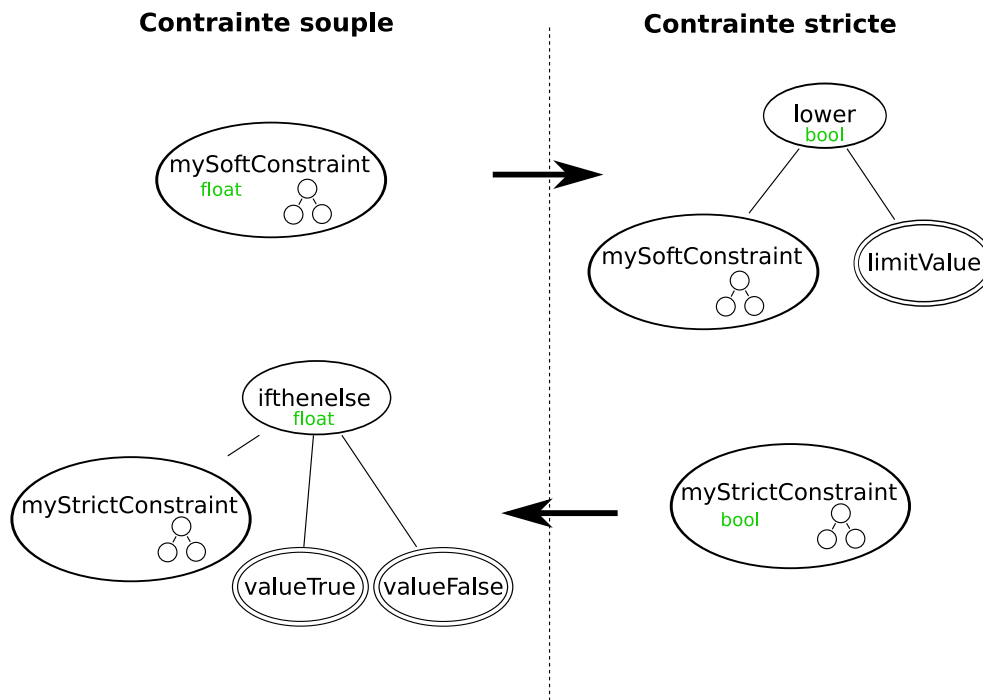


FIGURE 3.3: **Conversion contraintes strictes / contraintes souples** : le noeud composite *mySoftConstraint* représentant l'opération géométrique associée à une contrainte souple est de type flottant. Pour transformer la nature de la contrainte définie par l'opération géométrique il suffit de définir un seuil (*limitValue*) et d'utiliser l'opérateur *lower* dont le type est booléen. De même pour transformer une contrainte stricte dont le type du terme est booléen, en contrainte souple numérique, il faut utiliser l'opérateur *ifthenelse* qui affecte des valeurs numériques associées aux valeurs booléenne.

Dans le domaine de l'optimisation sous contraintes on parle de fonctions objectif et de contraintes (cf section 2.2.1). En réalité, la séparation entre les fonctions objectif et les contraintes peut paraître artificielle car une contrainte pourrait être considérée comme un objectif à atteindre. Cependant, elle se justifie d'une part parce que les contraintes sont appliquées sur l'espace de recherche alors que les objectifs définissent un ordre sur les solutions. D'autre part les contraintes et les objectifs sont généralement traitées par des procédures différentes. Nous avons donc choisi de les désigner sous le terme commun et général de *contrainte*, tout en distinguant leur nature par l'adjectif *souple* ou *stricte* selon le cas.

### 3.2.3 L'introduction de données complémentaires

Les données de base manipulées pour la définition des différentes contraintes sont constituées de l'anatomie du patient reconstruite en 3D et étiquetée à partir des images médicales. Les contraintes ne sont généralement pas exprimées simplement en fonction de l'anatomie spécifique de chaque patient. Nous avons introduit la possibilité d'ajouter de la connaissance aux données de base de façon à pouvoir manipuler des éléments complémentaires. Un troisième type de définition a donc été introduit pour faciliter l'expression des contraintes à respecter, il s'agit des *données complémentaires*. Une *donnée complémentaire* est un élément de type quelconque dont la valeur est déterminée par la racine du terme associé et auquel on affecte un nom par lequel il sera désigné dans la suite de la formalisation. Ces données ne sont pas des contraintes, mais leur fonctionnement et leur définition est similaire à celui des contraintes strictes et souples. Différents cas de figure ont motivé l'introduction de données complémentaires dans la formalisation des contraintes :

- on souhaite apporter des informations supplémentaires sur la nature des objets 3D constituant la scène. Nous avons à l'heure actuelle la possibilité de préciser si un organe est considéré comme un obstacle pour la suite des contraintes. On pourrait également imaginer par la suite pouvoir préciser d'autres propriétés comme par exemple le taux de perfusion d'un organe si cela peut être utile à la simulation du traitement.
- on souhaite que l'utilisateur du logiciel puisse donner des informations complémentaires sur le cas qui nous intéresse. Par exemple, le choix d'une stratégie peut être conditionné par l'âge du patient, ou par le fait que le foie auquel on s'intéresse est cirrhotique. Les données complémentaires permettent alors d'utiliser ces informations.
- l'utilisateur devrait également pouvoir intervenir sur certaines parties des contraintes pour les ajuster si nécessaire. Par exemple, si la contrainte à optimiser est une combinaison linéaire de plusieurs contraintes, l'utilisateur pourrait vouloir modifier les poids de ces contraintes de façon à obtenir une proposition plus proche de ce qu'il attend.
- il est possible qu'un élément soit utilisé plusieurs fois dans la définition des contraintes. Le définir en tant que donnée permet une meilleure lisibilité et compréhension de l'expression des contraintes. D'autre part cela entraîne une optimisation des calculs résultant puisque le terme correspondant sera évalué une fois pour toute au lieu de faire des opérations redondantes.

L'enrichissement de l'environnement de base peut donc se faire de différentes manières : soit en rajoutant directement de l'information, soit en la construisant à partir des données prédéfinies, soit en permettant à l'utilisateur de les renseigner. Toutes ces façons de rajouter de l'information sont regroupées dans la définition d'une donnée supplémentaire.



### 3.3 Définition de contraintes grâce au formalisme XML

Nous avons présenté les différents outils nécessaires à la définition d'une opération géométrique pouvant représenter différents types de contraintes. La définition d'une suite d'opérations géométriques n'a pas de sens en tant que telle. Il est nécessaire de structurer nos opérations de façon à ce que le rôle de chaque opération géométrique soit clairement défini en l'associant à une contrainte particulière dotée d'un certain nombre d'informations la décrivant. Le langage XML est le langage de référence pour l'échange des données et présente l'avantage d'être facilement reconnu et utilisable par le plus grand nombre. Nous avons donc choisi d'utiliser le système de balises et d'attributs XML pour structurer nos données. Le fichier XML qui regroupe les contraintes à respecter pour la planification de l'opération est divisé en trois parties. La première partie concerne les données complémentaires, la seconde les contraintes strictes et la troisième les contraintes souples. Ces différentes parties sont distinguées grâce aux balises XML `< data_definition >`, `< strict_constraints_definition >` et `< soft_constraints_definition >`. La grammaire détaillée du langage se trouve en annexe.

De manière générale, les contraintes sont exprimées sous forme d'une opération géométrique représentée par un *terme* tel que nous l'avons présenté dans la section . Le terme représentant une contrainte est encadré par une balise XML indiquant le type de contrainte le concernant, ainsi qu'un certain nombre d'informations annexes sur cette contrainte. Prenons un exemple simple :

$$\begin{aligned}
 & \langle \textit{soft\_constraint} \textit{name} = \textit{"volume constraint"} \textit{label} = \textit{"sc_vol"} \rangle \\
 & \textit{volume}(\textit{toolEffect}) \\
 & \langle \backslash \textit{soft\_constraint} \rangle
 \end{aligned}
 \tag{3.4}$$

Cette expression signifie que l'on définit une contrainte souple à optimiser que l'on appelle "volume constraint" dont le terme à optimiser est  $\textit{volume}(\textit{toolEffect})$ , c'est à dire le volume de la nécrose induite par le traitement. Les balises XML encadrent l'expression du terme représentant la contrainte. Un certain nombre d'attributs sont définis, permettant de spécifier des informations complémentaires sur la contrainte en cours de définition. Certains de ces attributs sont communs à toutes les contraintes (*name* et *label*), mais la plupart d'entre eux sont spécifiques au type de contrainte défini (voir le tableau 3.2 pour le récapitulatif des attributs existants). Les attributs *name* et *label* peuvent sembler redondants mais remplissent en réalité un rôle différent. Alors que *label* doit renseigner un nom relativement court pour pouvoir désigner la contrainte dans la suite du fichier du contrainte, *name* est destiné à décrire la contrainte pour l'utilisateur à travers l'interface de façon à ce que celui-ci comprenne facilement de quoi il s'agit.

Bien que les termes représentent des opérations géométriques représentables sous forme d'arbre (voir section 3.1.3), ils ne sont pas représentés directement en XML. En effet le formalisme XML permet de décrire très naturellement des structures de type arbre, cependant l'écriture sous forme fonctionnelle permet une écriture beaucoup plus concise et lisible.



TABLE 3.2: Description des attributs XML

Nom de l'attribut	Balises concernées	Valeur par défaut	Description de l'attribut
<i>name</i>	<i>data</i> , <i>strict_constraint</i> et <i>soft_constraint</i>	""	Nom explicite de la contrainte
<i>label</i>	<i>data</i> , <i>strict_constraint</i> et <i>soft_constraint</i>	""	Etiquette courte permettant la réutilisation du terme dans la suite des définitions
<i>show</i>	<i>data</i>	"false"	Booléen indiquant si la valeur de cette donnée est modifiable grâce à l'interface. Ce champ n'est pas applicable aux données dont le terme représente une forme.
<i>obstacle</i>	<i>data</i>	"true"	Booléen indiquant si la forme concernée par le terme doit être considérée comme un obstacle par l'opérateur <i>visible</i>
<i>agregate</i>	<i>soft_constraint</i>	"false"	Booléen indiquant si la contrainte décrite est la contrainte d'agrégation décrivant la fonction objectif
<i>mapped</i>	<i>soft_constraint</i>	"false"	Booléen indiquant si la contrainte souple est visualisable grâce à une carte colorée
<i>minvalue</i> , <i>maxvalue</i>	<i>soft_constraint</i>	"0", "1"	Valeurs de la contrainte correspondant à vert (resp. rouge) pour la cartographie de la contrainte. Attributs utilisés uniquement si <i>mapped</i> vaut "true".

### 3.3.1 Spécificité des données complémentaires

Nous avons évoqué dans la section 3.2.3 le rôle des données complémentaires dans la définition des contraintes. Syntaxiquement, la définition d'une donnée est similaire à la définition d'une contrainte qu'elle soit stricte ou non. Cette définition est constituée d'une balise XML annonçant le début de la déclaration (`< data... >`) dotée d'un certain nombre d'attributs. Puis intervient le terme définissant la valeur de la donnée et la fermeture de la balise (`< \data >`). Tout comme les données de base, les données complémentaires ainsi créées ont un type. Le type d'une donnée n'est pas explicitement précisé dans sa définition, cependant il se déduit du terme associé : le type d'une donnée est le type du noeud racine de son terme. Une fois définie, une donnée peut être réutilisée en tant que feuille d'un terme dans une déclaration ultérieure comme toute autre donnée de base.

Contrairement aux contraintes, la valeur d'une donnée dépend uniquement des données de base ou complémentaires précédemment définies. La modification de la stratégie

courante ne doit pas interférer avec la valeur de la donnée. C'est pourquoi les opérateurs spécifiques aux stratégies ne peuvent pas intervenir dans la définition du terme d'une donnée.

L'attribut *show* spécifique aux données permet de rendre la valeur de la donnée accessible et modifiable à l'utilisateur à travers l'interface. Par défaut, la valeur d'une donnée correspond à la valeur du noeud racine du terme une fois l'opération résolue. Cependant dans le cas où l'attribut *show* est mis à *true*, la valeur entrée par l'utilisateur écrase la valeur correspondant à l'évaluation du terme. Notons que la possibilité de mettre un point d'accès pour l'utilisateur est réservée aux objets de type simple : *bool*, *float* et *Point*. Prenons un exemple de définition de donnée :

$$\begin{aligned} &< \text{data name} = \text{"needle length"} \text{ label} = \text{"needleLength"} \text{ show} = \text{"true"} > \\ &150 \\ &< \backslash \text{data} > \end{aligned} \tag{3.5}$$

La donnée que l'on vient de définir est la longueur de l'aiguille. Cette donnée n'a pas de signification particulière pour le solveur, c'est simplement un réel qui vaut 150 et qui est réutilisable dans la suite des définitions. Cependant pour l'utilisateur qui aura accès à l'interface le nom de "needle length" aura une signification et il pourra lui affecter la valeur qu'il souhaite.

### 3.3.2 Reformulation du problème défini par les contraintes strictes

La définition d'une stratégie solution est liée à un ensemble de contraintes strictes décrit par l'expert (cf section 3.2.1). Si l'on cherche à déterminer si une stratégie est ou non une solution, il est *a priori* nécessaire de vérifier la validité des contraintes pour cette stratégie. Cette approche entraîne toutefois des inconvénients. D'une part, la vérification d'une stratégie est coûteuse en temps puisque, à chaque fois, toutes les contraintes strictes doivent être évaluées. D'autre part, cette façon de traiter les solutions ne permet pas d'avoir une vision globale et intuitive sur l'ensemble des solutions. Nous avons donc redéfini le problème de résolution des contraintes strictes sous la forme d'un problème plus générique pour lequel il n'y a en réalité que 3 contraintes strictes à observer :

- la pointe de l'outil est à l'intérieur du volume cible
- la trajectoire de l'outil traverse la zone d'insertion
- la zone d'effet de l'outil couvre tout le volume à traiter.

Cette redéfinition du problème nous amène à observer les stratégies potentielles sur les trois aspects suivants : la localisation de la pointe de l'outil, la localisation du point d'insertion de l'outil dans le corps du patient et la forme de la zone d'effet résultante. Les trois ensemble de délimitation d'aspect que sont le volume cible (délimitant la pointe), la zone d'insertion (délimitant le point d'insertion) et le volume à traiter (délimitant la zone d'effet minimale) déterminent le problème reformulé. En supposant qu'une solution du problème reformulé soit également une solution du problème initialement décrit par les contraintes strictes, il suffit alors de vérifier ces trois contraintes simples pour déterminer le caractère

solution d'une stratégie. Du point de vue du solveur, résoudre les contraintes strictes revient alors à calculer les ensembles de délimitation de façon la moins restrictive possible tout en garantissant que les solutions du problème reformulé sont des solutions du problème initial. Il n'est par contre pas possible de calculer des ensembles de délimitation pour lesquels les deux formulations du problème sont équivalentes dans le cas général. L'espace des solution résultant du problème reformulé est un sous-ensemble de l'espace des solutions du problème initial, il existe donc des stratégies solutions occultées par notre reformulation du problème. Notamment cette formulation oblige à restreindre l'espace dans lequel la pointe de l'aiguille est placée, et à ce que l'outil soit inséré à travers le patient. Cela ne semble pas pénalisant outre mesure puisque le praticien a une bonne connaissance *a priori* de la zone approximative qu'il souhaite cibler, et qu'une stratégie dont la trajectoire est externe au patient ne paraît pas vraisemblable. Néanmoins, il subsiste quelques limitations dues à cette reformulation, nous y revenons par la suite lorsque nous décrivons notre approche de résolution des contraintes par le solveur (section 4.3.2). A travers cette définition du problème, une fois les ensembles de délimitation calculés, la détermination des solutions est aisée et quasi immédiate. Par ailleurs, alors que l'espace des solutions du problème initial constitue un sous-ensemble de  $R^6$  difficilement manipulable, de cette façon, nous disposons d'ensembles représentables dans l'espace pour décrire l'espace des solutions. Les ensembles de délimitation ont un sens concret pour le praticien : où doit-il insérer son outil chirurgical, quel volume doit-il cibler avec son outil, et quel volume doit-il traiter pour que le respect des contraintes strictes soit garanti. La figure 3.4 illustre le rôle des ensembles de délimitation et la détermination des solutions résultantes de la reformulation du problème.

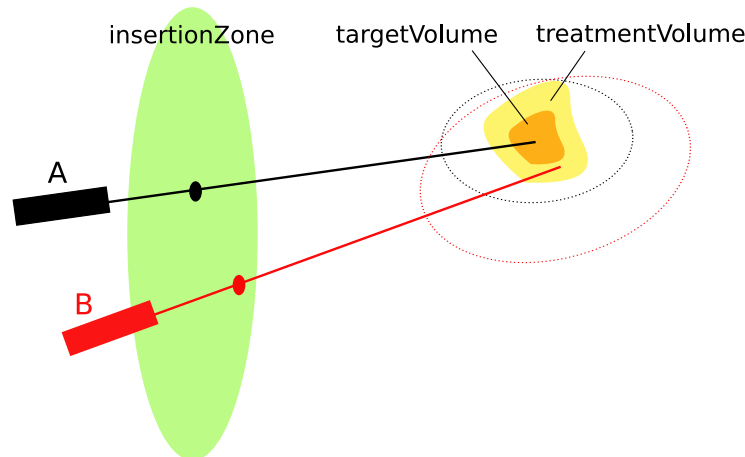


FIGURE 3.4: **Exemple de stratégie solution après redéfinition du problème** : La trajectoire A est une solution envisageable, la pointe de l'aiguille est placée dans le volume cible, elle traverse la zone d'insertion et la zone de nécrose englobe tout le volume à détruire. Au contraire, dans le cas de la trajectoire B la pointe de l'aiguille n'est pas placée dans le volume cible, cette possibilité n'est donc pas une solution.

### 3.3.2.1 Définition de termes correspondant à des contraintes strictes

Les ensembles de délimitation constituent des primitives du langage désignés par les noms *targetVolume*, *insertionZone* et *treatmentVolume*. Chaque contrainte définie dans le langage a un effet sur la construction de ces ensembles. Il est nécessaire que l'ensemble des contraintes définies par l'expert permettent de reformuler le problème sous la forme des trois contraintes interne utilisant les ensembles de délimitation. Pour cela nous avons défini deux types de contraintes :

- les contraintes *initiales* qui donnent une valeur initiale aux trois éléments de délimitation (ex : "la zone à traiter comprend la tumeur accompagnée d'une marge de 5 mm").
- les contraintes *restrictives* qui font intervenir une partie ou la totalité des paramètres et dont la résolution aboutit à la restriction des ensembles de délimitation de façon à garantir le respect de la contrainte à l'intérieur de l'espace des solutions résultant (ex : "l'aiguille ne peut pas traverser les os du patient").

L'initialisation des trois ensembles de délimitations qui déterminent l'espace des solutions est indispensable puisqu'elle est spécifique au type d'intervention considéré. Il est donc obligatoire de définir au moins une contrainte initiale pour chacun d'entre eux. Ces contraintes se reconnaissent grâce à l'utilisation des opérateurs *cross*, *target* et *cover* à la racine de l'opération les définissant. L'opérateur *cross* produit la donnée *insertionZone* qui est réutilisable comme n'importe quelle donnée dans la suite des définitions de contraintes. La valeur de cette donnée est actualisée à chaque utilisation de l'opérateur. Il en est de même pour les opérateurs *target* et *cover* qui produisent respectivement les données *targetVolume* et *treatmentVolume*.

Nous allons illustrer la définition des contraintes strictes *initiales* sur l'exemple de la radiofréquence hépatique. Sans tenir compte de contraintes élaborées, on peut déjà décrire la stratégie d'insertion d'outil de la façon suivante : l'aiguille doit traverser la peau, être placée dans la tumeur, et la zone de nécrose correspondante doit couvrir toute la tumeur accompagnée d'une marge de sécurité. Ces premières contraintes strictes peuvent être traduites sous forme de trois termes :

- `cross(skin)`
- `target(tumor)`
- `cover(enlarge(tumor,margin))`

Notons également qu'il est possible de définir plusieurs contraintes initiales pour le même élément. Par exemple, on aurait pu définir la contrainte d'initialisation de *treatmentVolume* en deux étapes : `cover( tumor )` puis `cover( enlarge(tumor, treatmentVolume) )`. Cela revient au même que ce que nous avons proposé en premier, cependant cela présente l'avantage de pouvoir intercaler une autre contrainte entre les deux et donc d'intervenir sur l'ordre dans lesquelles celles-ci vont être appliquées. Nous détaillerons les choix de formalisation auxquels on peut être confronté dans la section où nous présenterons notre fichier de contraintes.

Une fois que les contraintes définissant l'espace des solutions initial sont formalisées, il est possible de définir des contraintes strictes restrictives. Ce type de contrainte stricte est

défini par une opération dont la racine est booléenne et qui contient au moins un opérateur spécifique à la stratégie, rendant ainsi la valeur du terme variable, dépendant de la valeur de la stratégie courante. Prenons l'exemple de la contrainte suivante exprimée en langage naturel :

*"le point d'insertion de l'aiguille doit être suffisamment proche de la cible pour que celle-ci soit atteignable"*

Toujours en langage naturel mais en orientant la formulation selon les opérateurs géométriques impliqués, cette contrainte équivaut à la suivante :

*"la distance entre le centre de la cible et le point d'insertion doit être inférieure à la longueur de l'aiguille"*

On peut alors la traduire sous la forme du terme suivant :

*lower(distmin(center(targetVolume), toolInsertionPoint), needleLength)*

Cette contrainte signifie que l'on souhaite éliminer des solutions potentielles toutes les stratégies dont le point d'insertion est trop éloigné du centre de la cible. Au niveau de la résolution cela se traduit pas la réduction de l'ensemble *insertionZone* de façon à ce que tous les points d'insertion potentiels de la zone implique le respect de la contrainte (cf chapitre suivant, section 4.4.2).

### 3.3.3 Contraintes souples et optimisation

Les contraintes souples correspondent aux fonctions objectif de notre problème d'optimisation multi-critères. La valeur de chaque contrainte souple est numérique, dépend des variables de décision (regroupées dans la stratégie courante) et sa valeur est décroissante avec l'augmentation du respect de celle-ci (tous les problèmes d'optimisation peuvent se rapporter à un problème de minimisation). Pour que la contrainte soit valide, le terme représentant la contrainte souple doit donc représenter une fonction  $f : Strategy \rightarrow float$ . Cela signifie que ce terme contient au moins un opérateur manipulant la stratégie courante et que le type du noeud racine est réel.

On peut définir un nombre quelconque de contraintes souples, cependant comme nous l'avons évoqué à la section 2.3.4, nous avons choisi de résoudre notre problème d'optimisation multi-critère de manière agrégative, ce qui signifie qu'une seule contrainte sera la fonction objectif en définitive. Nous avons décidé de laisser la liberté de la méthode d'agrégation dans la description des contraintes. Il est alors nécessaire de créer une contrainte souple particulière dont le terme décrit comment les autres contraintes sont combinées entre elles pour former la fonction objectif. Les possibilités d'association grâce aux différents opérateurs disponibles sont multiples. Une possibilité est de former une combinaison linéaire des différentes contraintes, mais il est tout à fait possible de créer une fonction objectif plus compliquée. Il aurait été possible de ne pas intégrer la définition de la contrainte souple d'agrégation dans la définition des contraintes souples et de fusionner automatiquement

les différentes contraintes définies dans une moyenne pondérée dont chacun des éléments auraient un poids réglable. Cependant cette approche bride les possibilités offertes par notre langage de définition de contrainte pour gagner un peu en simplicité dans la définition des contraintes en supprimant la définition de la contrainte d'agrégation qui peut être relativement simple. La contrainte d'agrégation se distingue des autres grâce à l'attribut *aggregate* qui est mis à la valeur *true*. Une seule contrainte d'agrégation peut et doit être définie.

Une dernière spécificité de la définition des contraintes souples concerne l'attribut *mapped*. Nous verrons par la suite (cf section 5.2.3) que nous proposons une visualisation globale du résultat d'une contrainte à optimiser dans une partie de l'espace des solutions sous forme de cartes colorées. Par défaut, la carte est générée pour la contrainte agrégative. Cependant il est également possible de visualiser indépendamment certaines contraintes et donc d'activer la carte correspondante (c'est-à-dire mettre l'attribut *mapped* à *true*). Les attributs *minValue* et *maxValue* sont liés à l'attribut *mapped*. Dans le cas où l'on souhaite cartographier la contrainte, il faut préciser une plage de valeurs à laquelle s'attendre pour la contrainte. Ces attributs permettent d'associer une valeur minimale à la couleur verte et maximale à la couleur rouge, de façon à ce que les couleurs correspondant aux valeurs atteintes soient assez contrastées et permettent de visualiser facilement les parties de l'espace des solutions intéressantes.

### 3.3.4 Conformité du fichier de contrainte

Les contraintes sont donc décrites dans un fichier de données dans un langage que nous venons de décrire. Dans le cas où la syntaxe du langage n'est pas respectée, le fichier de contraintes n'est tout simplement pas chargé. Dans le cas contraire, même si la syntaxe est respectée, il est possible que le fichier en question ne soit pas conforme à ce qu'on pourrait en attendre.

#### 3.3.4.1 Désactivation d'une contrainte dont le terme n'est pas reconnu

La principale source d'erreur possible est alors le terme représentant l'opération à effectuer pour résoudre la variable ou la contrainte concernée. Différents cas peuvent être observés :

- un noeud interne du terme ne correspond à aucun opérateur connu
- un noeud feuille ne correspond ni à une constante numérique ou booléenne, ni à une variable prédéfinie
- le prototype d'un opérateur d'un noeud interne ne correspond pas au type de ses noeuds fils

Dans le premier cas, la variable/contrainte concernée par le terme est tout simplement désactivée. Le second cas peut se produire alors même que le terme est tout à fait valide, dans le cas où l'étiquette du noeud feuille désigne le nom d'un organe. Le noeud non identifié est alors supposé être un organe et est considéré comme étant de type *Shape*. Sa valeur n'est pas décrite dans le noeud tant que les données patient ne sont pas chargées. Après

le chargement de ces données patient, les valeurs des noeuds vides sont renseignées par les formes correspondant aux organes du même nom. C'est seulement lors de la première évaluation du terme que le solveur détectera si tous les noeuds feuilles ont une valeur définie. Si ce n'est pas le cas, la contrainte concernée est désactivée. Le troisième cas peut être vérifié directement après l'interprétation du fichier de contrainte, par un parcours en profondeur de l'arbre représentant le terme. Chaque noeud interne interroge ses fils sur leur type et vérifie qu'ils correspondent au prototype de son opérateur, puis il demande à ses fils d'en faire de même. Si une anomalie est détectée lors de cette vérification, la contrainte est désactivée.

La désactivation d'une contrainte entraîne différentes conséquences selon la nature de celle-ci. Tout d'abord, quelle que soit sa nature, la contrainte ne sera pas résolue lors du processus de planification. De plus la désactivation est signalée de façon à ce que la personne ayant décrit les contraintes soit informée que certaines d'entre elles ne sont pas utilisables de la façon dont elles ont été décrites. Si la désactivation concerne une variable, toutes les contraintes qui l'utilisent seront également désactivées. Dans le cas d'une contrainte stricte, il n'y a aucune précaution supplémentaire à prendre puisqu'elles ne sont pas réutilisées en elles-mêmes. Elles ne sont simplement pas appliquées pour définir l'espace des solutions. Enfin dans le cas d'une contrainte souple lorsqu'elles sont réutilisées dans la définition d'une autre contrainte, le noeud composite est remplacé par un noeud feuille dont la valeur est nulle. La désactivation en cascade n'est pas effectuée pour les contraintes souples pour éviter que la contrainte agrégative représentant la fonction objectif du problème ne soit désactivée à la moindre incohérence du fichier, et donc pouvoir proposer une planification malgré une définition incomplète des contraintes.

#### 3.3.4.2 Fichier de contraintes minimal

Un nombre réduit de définitions est nécessaire dans le fichier de description des contraintes pour que la planification automatique soit possible. La définition suivante correspond à un exemple minimal :

```
<constraint_file>
  <data_definition>
</data_definition>

  <strict_constraints_definition>
    <strict_constraint>
      target( tumor )
    </strict_constraint>

    <strict_constraint>
      cross( skin )
    </strict_constraint>

    <strict_constraint>
```



```
        include( tumor )
    </strict_constraint>
</strict_constraints_definition>

<soft_constraints_definition>
    <soft_constraint aggregate="true">
        1
    </soft_constraint>
</soft_constraints_definition>
</constraint_file>
```

Aucune définition de variable n'est nécessaire. L'espace de solutions doit être défini ; pour cela il faut au moins une contrainte stricte indiquant quel est le volume cible, quel est le volume à traiter et d'où est inséré l'outil. Enfin il est nécessaire de définir une fonction objectif permettant de choisir parmi les solutions du problème. Pour cela il suffit de définir une contrainte souple dont l'attribut `aggregate` est à `true`. Dans le cas où une de ces conditions n'est pas remplie par le fichier de contrainte, celui-ci est considéré comme complètement invalide et la partie solveur du logiciel de planification est désactivée tant qu'un fichier de contraintes valide ne lui est pas proposé.

### 3.4 Cas de la planification d'une opération de RFA hépatique : notre fichier de contraintes

Nous l'avons évoqué au début de ce chapitre, nos travaux ont à l'origine pour objectif de proposer une planification automatique pour l'opération de radiofréquence des tumeurs hépatiques. Nous avons donc rassemblé les informations disponibles dans la littérature médicale (*cf* section 1.2.1.4) et discuté avec plusieurs spécialistes du domaine afin de dégager un certain nombre de contraintes qui semblent être récurrentes dans les analyses des différents praticiens. Ces contraintes ont été formalisées dans un fichier de description de contraintes grâce au langage que nous avons décrit précédemment dans ce chapitre. Dans cette section, nous allons présenter ces contraintes et le fichier XML qui en découle. La description de ce fichier a pour double objectif de proposer une solution de planification pour la RFA hépatique et d'illustrer ce qu'il est possible de formaliser grâce à notre langage de description.

#### 3.4.1 Les contraintes à formaliser

Nous allons présenter, en langage naturel, les contraintes que nous avons décidé de prendre en compte dans le cas de la radiofréquence hépatique. Les contraintes strictes à respecter sont les suivantes :

1. L'aiguille traverse la peau et sa pointe doit être placée dans la tumeur traitée.



2. Le volume détruit suite à l'intervention doit englober la totalité du volume de la tumeur accompagnée d'une marge de tissus sains autour de celle-ci d'au moins 5 mm. Comme nous l'avons déjà évoqué à la section 1.2.1.4, cette marge de sécurité permet d'augmenter les chances que la totalité des tissus cancéreux soit effectivement détruite malgré les imprécisions pouvant intervenir à différentes étapes de l'opération.
3. L'aiguille ne doit pas endommager les organes vitaux du patient ainsi que les vaisseaux principaux, et ne peut pas traverser les os. Les dommages provoqués par la perforation d'un organe non concerné par l'intervention peuvent impliquer des complications sérieuses qu'il est indispensable d'éviter si l'anatomie du patient le permet.
4. La longueur de l'aiguille doit permettre d'atteindre la tumeur à partir du point d'insertion sélectionné.
5. La trajectoire de l'aiguille ne doit pas être trop tangente à la surface du foie (angle de tangence fixé à  $20^\circ$  ou inférieur), de façon à ce que l'insertion de celle-ci dans le foie soit possible en pratique. En effet, la membrane du foie (également appelée capsule) est particulièrement lisse et la consistance du foie est plutôt molle. Si l'on planifie une insertion très tangente au foie, en pratique l'aiguille glissera sur sa surface, n'atteindra pas sa cible et risquera d'endommager les organes à proximité.
6. Lorsque la tumeur est proche de la capsule du foie, la trajectoire de l'aiguille doit inclure une portion de foie sain. Cette précaution est prise afin de pouvoir permettre la cautérisation du chemin lors du retrait de l'aiguille qui peut provoquer une brûlure de la paroi abdominale dans le cas d'un retrait trop direct. Cette cautérisation est nécessaire pour ne pas disperser des cellules cancéreuses non détruites sur le chemin de l'outil.

Parmi les placements d'aiguille respectant ces contraintes strictes on souhaite sélectionner celui qui répond au mieux aux contraintes souples suivantes :

1. Le volume de tissus sains nécrosés par le traitement doit être minimal. Malgré la capacité du foie à se régénérer, la destruction d'une trop grande proportion de la réserve hépatique peut conduire à une altération de la fonction du foie, cela est d'autant plus le cas lorsque l'on agit sur un foie malade (cirrhotique).
2. La trajectoire de l'aiguille doit être la plus directe possible. Les praticiens expérimentés parviennent à placer l'aiguille avec une grande précision, cependant une certaine déviation par rapport à la trajectoire prévue est inévitable et risque d'être d'autant plus grande que la trajectoire est profonde.
3. Les trajectoires frôlant les organes vitaux doivent être évitées si possible. Il est évident qu'une trajectoire frôlant le poumon par exemple sera évitée s'il y a une autre possibilité de qualité équivalente. Notons que certaines structures peuvent être frôlées avec moins de risques que d'autres. Nous avons fixé à 1cm la limite de distances entraînant une pénalisation pour les différents organes de l'abdomen (hors foie), 5 mm pour les gros vaisseaux sanguins. Au delà de cette distance, on estime que le risque est nul. Les trajectoires frôlant les os n'entraînent pas de risque particulier et ne sont donc pas pénalisées.

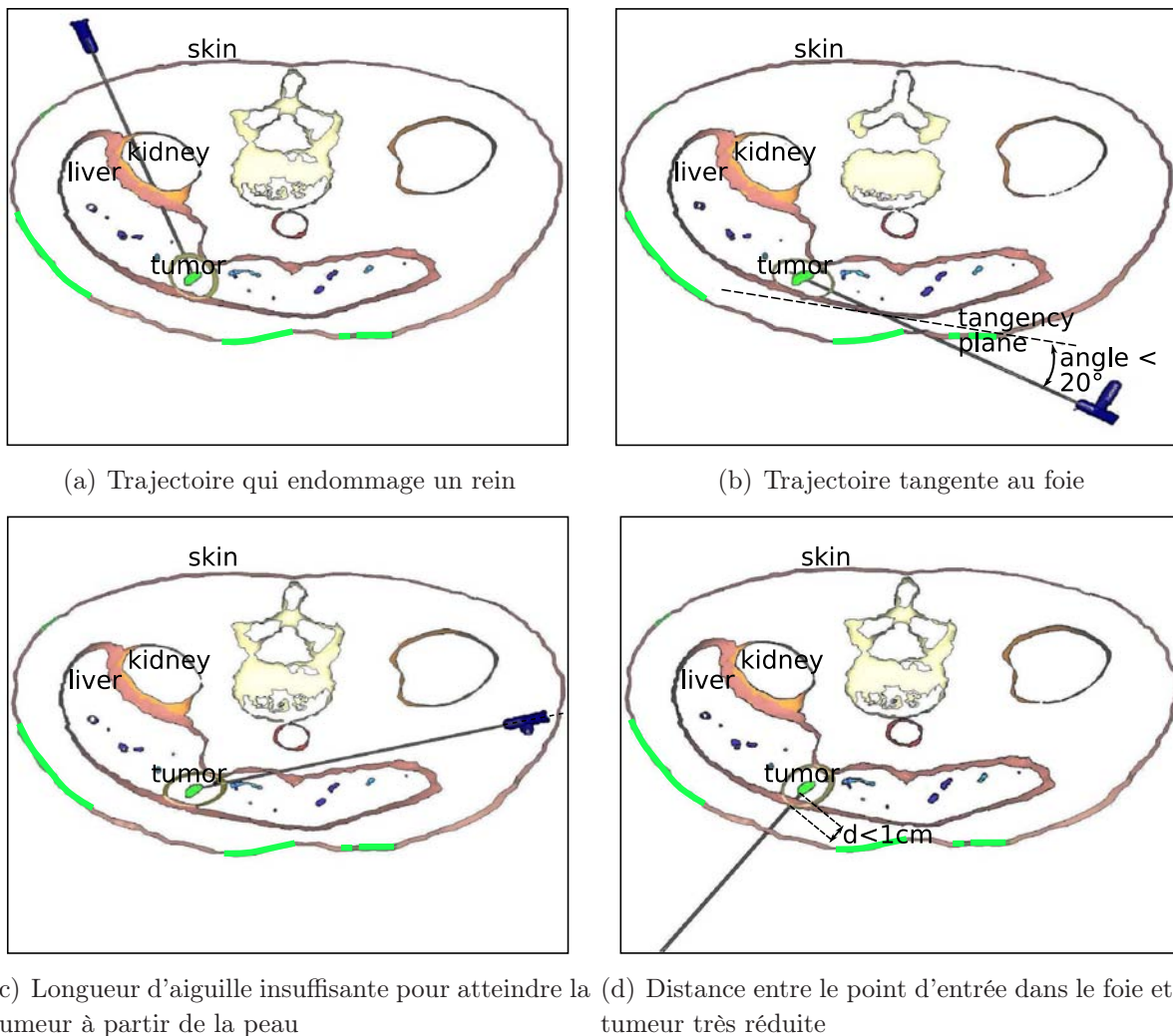


FIGURE 3.5: **Exemple de stratégies à éliminer de l'espace des solutions** : Dans chacune des coupes du patient virtuel, la stratégie représentée viole l'une ou l'autre des contraintes strictes. Elles n'appartiennent donc pas à l'espace des solutions représenté partiellement par la zone d'insertion sur la peau (en vert).

Les contraintes sélectionnées pour le cas de la RFA hépatique ont été traduites en opérations géométriques afin de pouvoir être interprétées par notre solveur de contraintes (fichier complet joint en annexe). Les possibilités d'exprimer ces différentes contraintes sont multiples. Nous allons donc présenter notre interprétation de ces contraintes en terme d'opérations géométriques et nous discuterons des raisons qui ont entraîné nos choix de formalisation. Pour cela nous allons tout d'abord étudier nos contraintes strictes, puis nous détailleront les contraintes souples. La définition de variables étant motivée la plupart du temps par la formalisation d'une contrainte stricte ou souple, celles-ci seront évoquées simultanément à la contrainte concernée.

### 3.4.2 Définition des contraintes strictes

#### 3.4.2.1 "L'aiguille traverse la peau et sa pointe doit être placée dans la tumeur traitée"

Ces deux contraintes définissent un espace de recherche initial pour la position de la pointe de l'outil ainsi que la position du point d'insertion. On utilise donc directement les opérateurs définis pour initialiser *insertionZone* et *targetVolume* :

```
<strict_constraint name= "initial target volume" label="str_1">
    target( tumor )
</strict_constraint>

<strict_constraint name= "initial insertion zone" label="str_3">
    cross( skin )
</strict_constraint>
```

#### 3.4.2.2 "Le volume traité doit englober la totalité du volume de la tumeur accompagnée d'une marge de 5 mm."

Cette contrainte initialise *treatmentVolume* qui représente le volume que l'outil doit englober dans sa zone d'effet. Elle est définie dans le fichier de contraintes en deux étapes. Tout d'abord on définit la donnée *tumor\_margin* représentant la forme issue de la dilatation de la tumeur de 5mm :

```
<data name="tumor and its margin" label="tumor_margin">
    enlarge( tumor, 5 )
</data>
```

Ensuite dans la définition des contraintes strictes, il suffit simplement d'affecter la valeur de la variable *tumor\_margin* au volume à traiter en utilisant l'opérateur *include*.

```
<strict_constraint name= "initial treatment volume" label="str_2">
    include( tumor_margin )
</strict_constraint>
```

Cette contrainte pourrait très bien être définie en une seule étape avec le terme *include(enlarge(tumor,5))*. L'avantage de définir la tumeur dilatée en tant que donnée est de pouvoir la réutiliser sans la recalculer dans d'autres définitions.

#### 3.4.2.3 "L'aiguille ne doit pas endommager les organes vitaux du patient"

Nous avons décidé de modifier ce problème de collisions avec les organes en un problème de visibilité. Les raisons de cette approche seront explicitées dans le chapitre suivant lorsque nous présenterons l'opérateur *visible* développé spécifiquement pour cette contrainte (section 4.2). Les points de la zone d'insertion qui permettent d'atteindre le volume cible sans

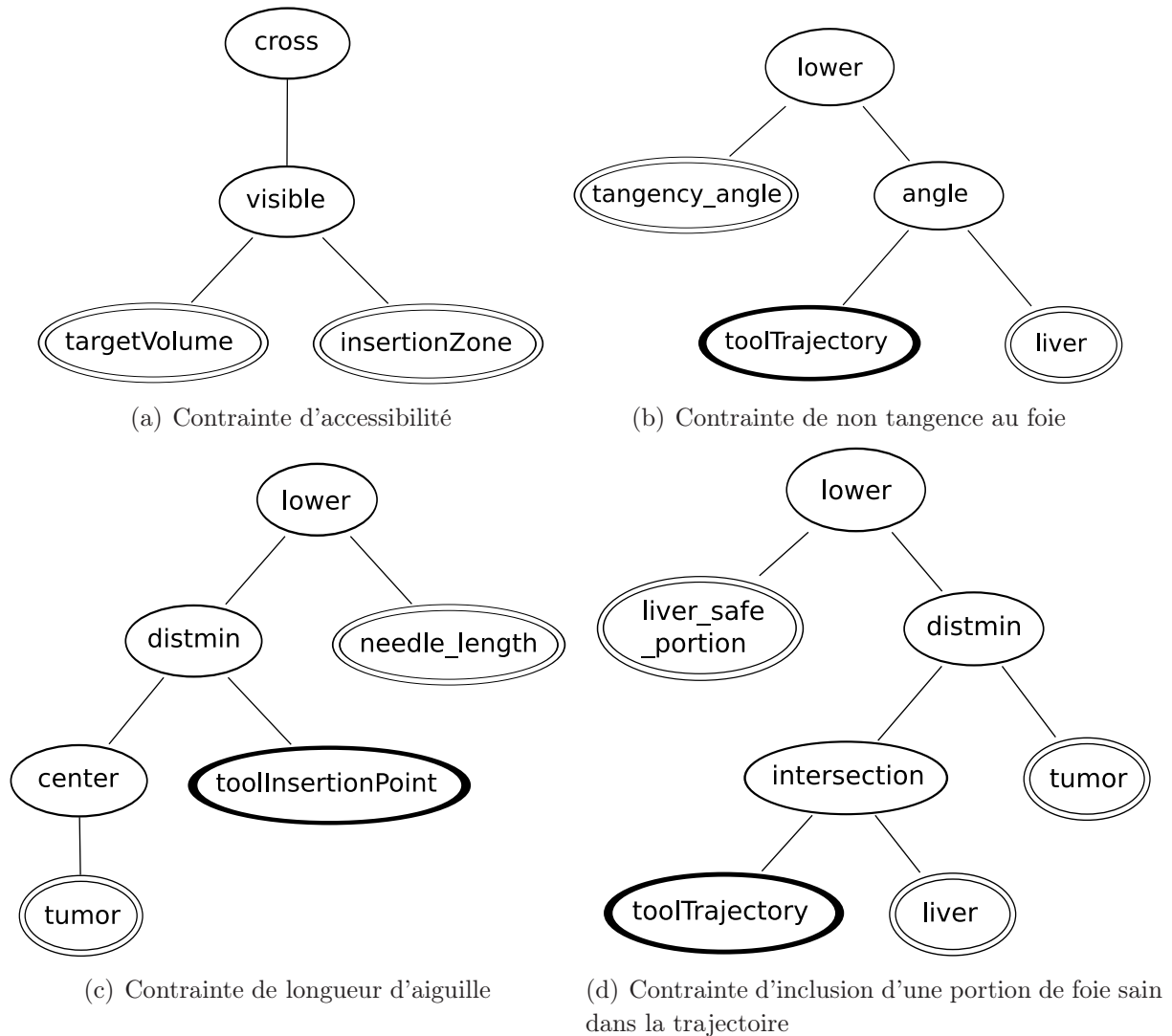


FIGURE 3.6: Opérations représentant les contraintes strictes sélectionnées

endommager les organes environnants sont ceux qui sont visibles depuis ce même volume en considérant tous les organes à éviter comme des obstacles occultants. Une stratégie respectant cette contrainte doit alors traverser la partie de *insertionZone* qui est visible depuis *targetVolume* (figure 3.6(a)) :

```
<strict_constraint name="accessibility" label="str_acces">
  cross(visible( insertionZone , targetVolume ))
</strict_constraint>
```

L'utilisation de l'opérateur *visible* est liée avec la caractéristique *obstacle* des objets de la scène. Par défaut toutes les *Formes* constituant la scène représentant l'anatomie du patient sont considérées comme des obstacles pour l'opérateur *visible*. Or le foie n'en est pas un puisqu'il faut nécessairement le traverser pour atteindre la tumeur hépatique. Ce n'est pas non plus le cas de la peau. Pour pouvoir changer le statut du foie ainsi que de la

peau, il est donc nécessaire de les redéfinir en tant que donnée complémentaire en modifiant l'attribut *obstacle* :

```
<data name="liver" label="liver" obstacle="false">
</data>
```

Notons que cette contrainte affecte une valeur à la zone d'insertion, en utilisant la zone d'insertion courante résultant de l'application des contraintes strictes précédentes. Nous aurions pu directement définir la contrainte initiale dont le terme associé est le suivant : `cross(visible(skin, targetVolume))`. Nous n'avons pas procédé de cette manière afin de pouvoir séparer la contrainte initiale qui définit la peau comme zone d'insertion initiale de la contrainte d'évitement d'obstacles, et ainsi pouvoir intercaler une contrainte entre les deux définitions.

#### 3.4.2.4 "La longueur de l'aiguille doit permettre d'atteindre la tumeur "

Cette contrainte signifie simplement que la distance entre la cible et le point d'insertion doit être inférieure à la taille de l'aiguille (voir figure 3.6(c)). Pour pouvoir évaluer cette contrainte nous avons sélectionné comme cible probable le centre de la tumeur. Nous aurions pu sélectionner tout le volume cible comme cible potentielle, cependant cela ne nous a pas semblé nécessaire puisque la cible est généralement proche du centre de la tumeur et que l'opérateur de distance entre un point et un autre point est moins coûteux que la distance d'un point à une forme. La contrainte est donc la suivante :

```
<strict_constraint name="needle size" label="str_needlesize">
  lower (
    distmin ( center( tumor ), toolInsertionPoint ),
    needle_length
  )
</strict_constraint>
```

#### 3.4.2.5 "La trajectoire de l'aiguille ne doit pas être trop tangente à la surface du foie"

Cette contrainte signifie que l'angle entre la trajectoire de l'aiguille et la surface du foie doit être supérieur à une valeur limite d'angle de tangence dont la valeur par défaut est fixée à 20°. Nous n'avons pour l'instant défini qu'un nombre limité d'opérateurs, l'opération de comparaison s'effectue donc avec l'opérateur *lower* ce qui implique l'inversion de l'opération par rapport à cette formulation (voir figure 3.6(b)) :

```
<strict_constraint name="tangency" label="str_tangency">
  lower (
    tangency_angle ,
    angle( toolTrajectory , liver )
  )
```

```
</strict_constraint>
```

### 3.4.2.6 "La trajectoire de l'aiguille doit inclure une portion de foie sain"

Cette contrainte n'est éliminatoire pour certaines trajectoires que lorsque la tumeur est proche de la capsule puisque l'ablation d'une tumeur située plus en profondeur dans le foie implique nécessairement d'inclure une portion de foie sain dans la trajectoire quel que soit l'angle d'attaque. Cela explique pourquoi les praticiens ont jugé nécessaire de préciser que cette contrainte s'applique uniquement pour les tumeurs sous-capsulaires, alors qu'en réalité il n'est pas nécessaire de conditionner l'application de cette contrainte à une localisation de tumeur particulière. Il s'agit alors de déterminer si l'intersection entre la trajectoire de l'outil et le foie est suffisamment distante de la tumeur (voir figure 3.6(d)) :

```
<strict_constraint name="liver_portion" label="str_liverportion">
  lower(
    liver_safe_portion ,
    distmin( intersection( toolTrajectory , liver ) ,
             tumor )
  )
</strict_constraint>
```

### 3.4.3 Les contraintes strictes : choix de formalisation

Les contraintes strictes que nous venons de définir entraînent différentes observations. La première remarque concerne la nature des contraintes strictes. Nous avons déjà évoqué plus tôt que les contraintes strictes sont de nature booléenne, c'est-à-dire qu'elles sont soit respectées soit enfreintes. Cependant une expression de contrainte numérique n'implique pas forcément une contrainte souple. Cette constatation est illustrée par les contraintes strictes que nous venons de présenter. Par exemple, la contrainte qui concerne l'angle d'insertion dans le foie par exemple est de nature numérique. Pourtant lorsque nous l'avons évoquée avec le praticien, celui-ci a évoqué les trajectoires tangentes au foie comme des trajectoires à éliminer. C'est ce caractère éliminatoire de la contrainte qui implique la nature stricte de la contrainte.

Les contraintes d'inclusion d'une portion de foie sain et de profondeur d'insertion limitée par la taille de l'aiguille fonctionnent de la même manière. La transformation de ces contraintes à valeur numérique en contraintes booléennes implique la définition d'un seuil départageant les valeurs correspondant aux valeurs booléennes *true* et *false*. Pour chacune de ces contraintes, nous aurions pu fixer ce seuil à une valeur déduite de nos entretiens, ou de la bibliographie médicale. Cependant ces seuils nous paraissent être typiquement des paramètres spécifiques au praticien qui pratique l'intervention et doivent pouvoir être modifiés aisément par celui-ci sans intervenir dans le fichier de contraintes. C'est pourquoi nous avons défini une donnée pour chacun de ces seuils. Par exemple :

```
<data name="portion of safe liver tissues crossed "
label="liver_safe_portion" show="true">
    10
</data>
```

Cette donnée, dont la valeur par défaut est fixée à 10 mm, sera alors accessible à l'utilisateur du solveur à travers l'interface sous le nom "portion of safe liver tissue crossed" grâce à l'attribut *show* positionné à *true*. Dans le fichier de contraintes la donnée est utilisable en tant que noeud feuille d'un terme grâce à son label *liver\_safe\_portion*.

Enfin évoquons l'ordre de définition des contraintes strictes restrictives. A priori cet ordre n'a pas d'importance sur l'espace des solutions résultant de leur interprétation puisque les contraintes strictes doivent toutes être respectées. Cet ordre a cependant un impact sur l'ordre dans lequel les filtres correspondant aux contraintes sont appliqués sur l'espace des solutions initial. Nous avons donc choisi de commencer par les contraintes les moins coûteuses en terme de temps de calcul, de façon à ce qu'elles éliminent un maximum de solution qu'il ne sera pas nécessaire de tester pour les contraintes qui entraînent un calcul plus lourd. En pratique, sur les exemples de patients virtuels que nous avons recueillis, le changement d'ordre des contraintes strictes a cependant peu d'incidence sur le temps d'exécution qui est inférieur à la minute dans la plupart des cas.

### 3.4.4 Définition des contraintes souples

#### 3.4.4.1 "Le volume de tissus sains nécrosés par le traitement doit être minimal"

```
<soft_constraint name= "volume constraint" label="soft_vol"
mapped="true">
    volume( toolEffect )
</soft_constraint>
```

Cette contrainte consiste simplement à minimiser le volume de la zone d'effet associée au placement.

#### 3.4.4.2 "La trajectoire de l'aiguille doit être la plus directe possible"

```
<soft_constraint name= "depth constraint" label="soft_depth"
mapped="true">
    distmin( toolInsertionPoint , toolTip )
</soft_constraint>
```

Une trajectoire directe signifie que la distance entre le placement de la pointe de l'outil et le point d'insertion est minimal.



### 3.4.4.3 "Les trajectoires frôlant les organes vitaux doivent être évitées"

La contrainte de risque due à la proximité des organes est définie en plusieurs étapes. Dans un premier temps on déclare une donnée représentant l'ensemble des organes à éviter :

```
<data name="vital_organ" label="organ">
  union( lung , heart , spleen , stomach , kidney ,
         pancreas , intestine )
</data>
```

Cela permet de créer un objet de type *Shape* utilisable par l'opérateur de distance entre objets géométriques. L'opérateur *union* est le seul opérateur qui ne déclenche pas d'exception entraînant la désactivation de la contrainte si l'une de ses opérands n'a pas de valeur valide. Il considère simplement que la forme en question est une forme vide. Ainsi même si les données patient chargées ne contiennent pas l'un ou l'autre organe, la planification peut tout de même être effectuée à partir des données disponibles. Il est alors possible de définir la contrainte de risque due à la proximité des organes de la façon suivante :

```
<soft_constraint name="risk_constraint_for_organ"
label="soft_risk_organ" mapped="false">
  max( minus( distmin( toolTrajectory , organ ) ),
        minus(risk_limit_organ) )
</soft_constraint>
```

Contrairement aux deux autres, la contrainte de risque sous-entend une maximisation de la distance entre la trajectoire et l'organe considéré. Tout problème d'optimisation peut être ramené à un problème de minimisation. Nous avons donc décidé que toutes les contraintes souples décrites sont à minimiser. Dans le cas de la maximisation, il suffit pour cela de considérer la fonction opposée de la fonction objectif. En considérant les contraintes souples sous forme de termes, cela revient à ajouter l'opérateur *minus* à la racine du terme correspondant. D'autre part, comme nous l'avons précisé dans la description des connaissances à formaliser, au delà d'une certaine distance le risque est nul et donc la valeur de la contrainte associée devrait être la même de façon à ce qu'une trajectoire qui passe à 10 cm de l'organe considéré ne soit pas privilégiée par rapport à une trajectoire éloignée de seulement 5 cm. Pour cela nous avons utilisé l'opérateur *max* permettant de fixer la valeur minimale de l'opération associée à la contrainte à - (*risk\_limit\_organ*). Une distance de sécurité moins large a été définie pour le système vasculaire du patient (5 mm). Les gros vaisseaux sanguins sont donc traités à part, dans leur propre contrainte souple.

```
<soft_constraint name="risk_constraint_for_vessels"
label="soft_risk_vessels" mapped="false">
  max( minus( distmin( toolTrajectory , vessels ) ),
        minus(risk_limit_vessels) )
</soft_constraint>
```



Les deux contraintes souples intermédiaires sont alors fusionnées dans une seule contrainte de risque et c'est cette contrainte qui sera accessible pour l'utilisateur à travers l'interface. Les deux sous-contraintes ayant leur attribut *mapped* à *false*, leur existence sera complètement transparente pour une utilisation simple à travers l'interface du logiciel. Seul l'expert qui définit les contraintes de planification en a connaissance.

```
<soft_constraint name="risk_constraint" label="soft_risk">
    plus( soft_risk_organ , soft_risk_vessels )
</soft_constraint>
```

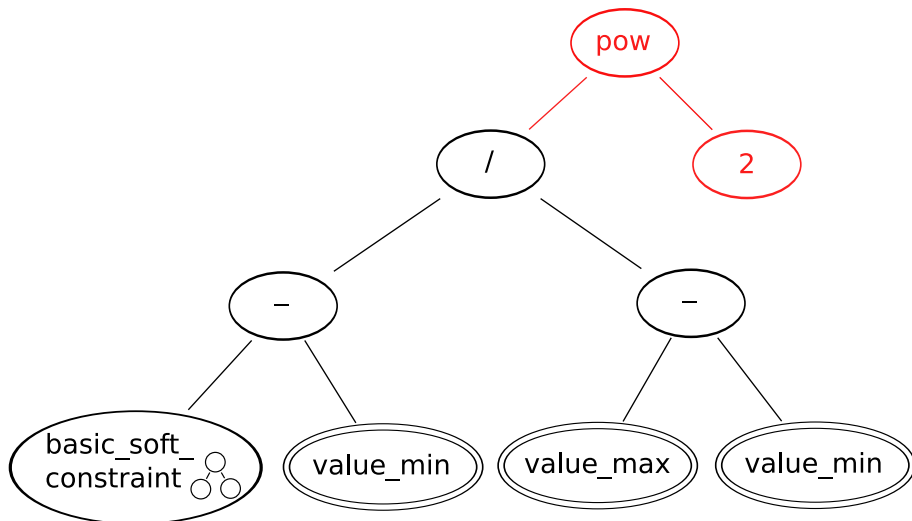
### 3.4.5 L'agrégation des contraintes souples

#### 3.4.5.1 Mise à l'échelle des différentes contraintes

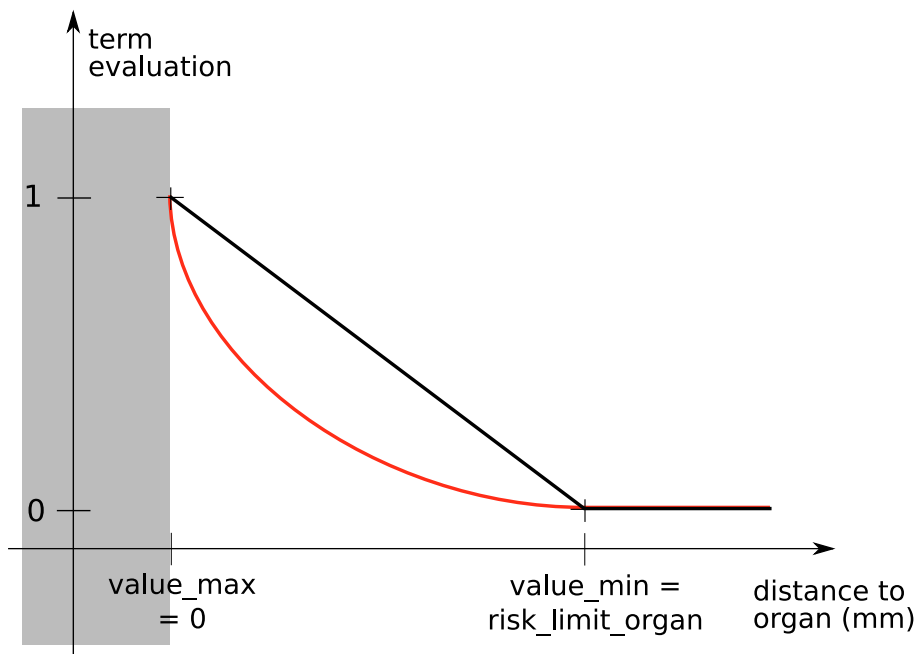
Nous venons de présenter les trois contraintes souples que nous avons sélectionnées sous forme de termes. Puisque nous avons décidé d'adopter l'approche d'agrégation de nos fonctions objectif pour tenir compte de nos multiples critères d'optimisation, il est nécessaire de les combiner dans un seul terme représentant la fonction objectif de notre problème d'optimisation. Cependant, cela n'a pas de sens d'agréger nos trois contraintes en une seule telles qu'elles ont été décrites en appliquant par exemple une moyenne pondérée. En effet la profondeur d'insertion est de l'ordre de plusieurs dizaines de millimètres, alors que le volume d'effet de l'outil se compte en milliers de millimètres cubes. Il est donc nécessaire d'effectuer une normalisation pour chacune des contraintes, de façon à ce que les valeurs obtenues soient d'ordre comparable et ne viennent pas parasiter le poids de la contrainte dans la moyenne pondérée représentant l'importance de la contrainte dans la prise de décision finale.

Les termes représentant les contraintes souples tels que nous venons de les présenter ne correspondent donc pas aux termes définitifs présentés dans notre fichier de contrainte. Ils représentent le coeur du terme associé à la contrainte avant la mise à l'échelle que nous allons décrire. Nous avons choisi de redimensionner les contraintes de façon à ce que la majorité des valeurs atteintes pour chacune d'elles se situe entre 0 et 1, 0 représentant donc les cas les plus favorables et 1 les cas les plus critiques vis-à-vis de la contrainte. La figure 3.7 représente le terme associé au redimensionnement d'une contrainte souple quelconque. Le redimensionnement est nécessairement associé à la définition de deux constantes représentant les valeurs des bornes. C'est-à-dire qu'il est nécessaire de définir une valeur de la contrainte pour laquelle celle-ci est totalement respectée (*value\_min*) et donc associée à une évaluation de 0 après mise à l'échelle, mais également une valeur correspondant à un non respect maximal de la contrainte (*value\_max*) et associée à une évaluation de 1.

Le redimensionnement de la contrainte ne pose pas simplement la question des valeurs limites mais aussi de la distribution des valeurs entre ces bornes. Il y a évidemment une infinité de façon de redimensionner une fonction entre 0 et 1. Ces différentes façons sont utilisables avec notre langage de description du moment que les opérateurs existants le permettent. Nous avons choisi ici d'illustrer les façons les plus simples, c'est-à-dire la dis-



(a) Construction du terme redimensionné



(b) Valeur de la contrainte de risque en fonction de la distance à l'organe considéré

**FIGURE 3.7: Redimensionnement d'une contrainte souple :** La figure a) représente la construction du terme redimensionné à des valeurs entre 0 et 1, dispersées de façon linéaire (terme en noir) ou de façon quadratique (terme avec la racine en rouge). La figure b) est la représentation graphique de notre fonction de mise à l'échelle de notre contrainte de base. L'exemple représenté est la contrainte de risque associée à la proximité d'un organe. La courbe en noir correspond à la version linéaire et celle en rouge à la version quadratique. La partie grisée représente les valeurs non atteignables.

tribution linéaire ainsi que la distribution quadratique. En pratique nous n'avons utilisé que la distribution linéaire des valeurs dans notre fichier de contraintes définitif, cepen-

dant la distribution quadratique ne paraît pas dénuée d'intérêt. En effet prenons comme exemple la contrainte de risque liée à la proximité d'un organe vital. On peut tout à fait imaginer que passer d'une trajectoire plus proche de 1 mm qu'une autre est plus pénalisant si la trajectoire est déjà à 2 mm de l'organe, par rapport à une trajectoire à 9 mm. Cela pourrait être représenté par exemple par un redimensionnement quadratique. Notre vision des contraintes intervenant dans le choix d'une intervention ne nous permettant pas de les définir de façon aussi précise, nous avons choisi de privilégier la simplicité dans notre fichier de contraintes définitif et donc la distribution linéaire.

Un dernier point de la mise à l'échelle des contraintes qui doit être discuté concerne les valeurs atteintes situées en dehors de l'intervalle  $[value\_min, value\_max]$ . Il y a alors plusieurs possibilités qui peuvent s'appliquer selon la contrainte concernée :

- les valeurs situées au delà de la borne ne sont pas atteignables ou ne sont pas atteintes dans l'espace des solutions. Nous pouvons citer deux exemples pour illustrer ces deux cas. Concernant la contrainte de risque il n'est tout simplement pas possible de trouver une trajectoire pour laquelle la distance à un organe sera inférieure à 0. Quant à la profondeur d'insertion, la valeur représentant le pire des cas correspond à la longueur de l'aiguille. Or nous avons défini une contrainte stricte retirant de l'espace des solutions toutes les trajectoires dont la profondeur est supérieure à la taille de l'aiguille. Ce cas-là est le plus simple puisqu'il n'est pas nécessaire de se soucier de ce que notre fonction de mise à l'échelle produit en dehors de l'intervalle (partie grisée sur la figure 3.7(b)).
- les valeurs situées en dehors de la borne sont atteignables mais doivent être considérées comme si elles étaient égales à la borne. C'est le cas de la contrainte de risque, lorsque la distance à un organe est supérieure à la distance limite que l'on a fixée à 1 cm. Le risque doit être minimal (c'est-à-dire nul) que l'on soit à 1cm, 5 cm ou 20 cm de l'organe considéré. On utilise alors l'opérateur *min* ou *max* selon le cas pour niveler les valeurs atteintes au delà des bornes.
- les valeurs au delà des bornes peuvent être atteintes, mais on souhaite quand même faire une différence entre elles. C'est le cas de la contrainte de volume de la nécrose. La valeur minimale pour cette contrainte est le volume de l'ellipsoïde englobant minimal de la tumeur, et la valeur maximale est fixée à 160% de ce même volume. Pour cette contrainte on ne sait pas trouver une borne supérieure, nous avons donc choisi une proportion du volume minimal qui correspond à la perte de volume qui semble le maximum acceptable. Le fait de dépasser cette valeur devient très pénalisant pour une stratégie qui sera écartée plus facilement. Le poids de la contrainte devient alors exacerbé lorsque l'on s'écarte des valeurs définies par les bornes. Il aurait été envisageable d'associer une contrainte stricte à cette contrainte souple, de façon à éliminer de l'espace des solutions les trajectoires qui posent problème. Cependant cela ne nous a pas paru pertinent, le volume de tissus détruit n'étant pas systématiquement éliminatoire dans le discours des praticiens.

Les valeurs des contraintes situées hors des bornes de mise à l'échelle ne nécessitent donc généralement pas de traitement particulier, si ce n'est, parfois, l'ajout d'un opérateur dans

la définition du terme de la contrainte. Une fois la mise à l'échelle effectuée et les valeurs extrêmes traitées spécifiquement à chaque contrainte comme nous venons de le décrire, la plupart des valeurs atteintes dans le domaine des solutions pour les différentes fonctions objectif appartiennent à l'intervalle  $[0,1]$ .

### 3.4.5.2 Contrainte représentant la fonction objectif

Pour résoudre notre problème d'optimisation multi-objectifs, nous avons choisi l'approche agrégative qui consiste à fusionner nos différentes fonctions objectif dans une seule et même fonction qui sera traitée comme s'il s'agissait d'un problème d'optimisation mono-objectif. Nous avons opéré une mise à l'échelle de ces fonctions objectif de façon à pouvoir manipuler des quantités comparables pour les différents critères considérés. Comme nous l'avons déjà évoqué, les possibilités d'agrégation sont multiples. Dans le fichier de contraintes que nous proposons, nous avons choisi une moyenne pondérée de nos trois contraintes souples :

```
<soft_constraint name="final" label="soft_final"
  aggregative="true" mapped="true" minValue="0" maxValue="1">
  divide(plus(
    mult( weight_sc_depth , sc_depth ),
    mult( weight_sc_vol , sc_vol ),
    mult( weight_sc_risk , sc_risk ) ),
    plus( weight_sc_depth , weight_sc_vol , weight_sc_risk ) )
</soft_constraint>
```

Le poids de chacune des contraintes est déclaré comme une donnée dont la valeur par défaut est de  $1/3$  :

```
<data name="weight of depth constraint"
  label="weight_soft_depth" show="true">
  divide(1/3)
</data>
```

Cette déclaration en tant que donnée permet à l'utilisateur du logiciel d'intervenir dans le poids de chacune des contraintes s'il le souhaite (attribut *show* à *true*). En effet, nos entretiens avec les spécialistes nous ont permis de déterminer ces différents critères intervenant dans la sélection d'une stratégie, cependant ils ne nous ont pas permis de définir avec précision leur importance respective. D'autant plus qu'il nous semble que ces différents poids peuvent varier d'un spécialiste à l'autre. Nous avons donc choisi de faire intervenir nos trois critères avec un poids équivalent par défaut mais réglable si nécessaire.

## 3.5 La formalisation des contraintes : conclusions et perspectives

Dans ce chapitre, nous avons présenté un langage de formalisation spécifique à la définition de contraintes pour le choix d'une stratégie d'opération de chirurgie mini-invasive. Les contraintes que nous avons décelées sont traduisibles sous forme de combinaisons d'opérations géométriques faisant intervenir les outils et les données du patient. Nous avons donc présenté les opérateurs et les données manipulables permettant de définir un terme définissant une contrainte. Nous avons alors évoqué les différentes natures des contraintes à formaliser et notamment les notions de contraintes strictes et souples intervenant spécifiquement dans le processus de résolution : les contraintes strictes définissent l'espace des solutions envisageables, les contraintes souples permettent d'effectuer un classement des stratégies solutions, afin que l'on puisse sélectionner la meilleure d'entre elles. Puis nous avons présenté le langage de définition de contraintes, inspiré du formalisme XML. Enfin nous avons proposé un fichier de contraintes spécifique au traitement des tumeurs hépatiques par radiofréquence percutanée, à titre d'exemple mais également suites aux nombreuses interactions avec les praticiens expérimentés de cette intervention.

A travers ce travail de formalisation des contraintes dans un langage spécifique, nous avons voulu nous rapprocher de l'approche décrite par la modélisation déclarative surtout utilisée dans le domaine de la CAO. La stratégie d'opération est alors décrite par un ensemble de propriétés à résoudre. Nous venons de présenter la phase de description de ces propriétés que nous avons désignées sous le terme de contraintes. Le langage spécifique à cette description est volontairement simple et accessible, cependant il est encore relativement formel. On pourrait envisager de proposer un langage plus proche du langage naturel, et notamment du langage et du vocabulaire technique du praticien de façon à ce qu'il soit envisageable que celui-ci définisse lui même les contraintes propres à son type d'opération, de la même façon que le concepteur définit les contraintes de la scène qu'il modélise. Ce travail nécessiterait des interactions fortes avec différents praticiens afin de déterminer un dictionnaire de termes adaptés ainsi que des tournures de phrases plus proches de celles qu'ils pourraient exprimer naturellement [LGD06].

D'autre part, même si le langage proposé dans ce chapitre a été conçu de façon à être accessible et que l'on envisage une approche plus proche du langage naturel dans le futur, il pourrait être intéressant de développer encore l'aspect intuitif de la définition des contraintes. La conception des règles de planification pourrait être effectuée à l'aide d'une interface multimodale. Cette interface pourrait inclure la visualisation de l'arbre représentant un terme en parallèle avec la définition d'une contrainte en langage naturel, sa manipulation directe, la construction facilitée de termes normalisés et la vérification de la cohérence des contraintes définies. Un tel outil permettrait de rendre envisageable la définition directe des contraintes par le praticien ne connaissant pas le formalisme XML.

Les règles de planification spécifiques à la radiofréquence ont été mises en évidence au cours de différents entretiens avec des praticiens expérimentés. Les contraintes décrites dans le fichier de contraintes spécifique à la radiofréquence ont été définies selon leurs in-

dications. Les praticiens nous ont confirmé leur validité et les ont enrichies au fur et à mesure des entrevues. Les résultats de ces contraintes sur quelques patients leur ont également été présentés et ont été approuvés. Nous aimerions effectuer une validation médicale plus complète et plus détaillée. Pour cela, il serait nécessaire de comparer le résultat de notre solveur avec la vérité terrain, c'est-à-dire dans notre cas les stratégies d'opérations effectivement optimales pour chaque cas considéré. Cependant la détermination de cette vérité terrain n'est pas une question évidente. En effet, on pourrait éventuellement envisager de comparer le résultat de la planification automatisée avec la stratégie mise en oeuvre en réalité. Toutefois, il est possible que la stratégie sélectionnée par le praticien ne corresponde pas à la solution optimale. On ne peut donc se contenter de comparer la stratégie proposée par le solveur avec la stratégie sélectionnée par le praticien. Il est nécessaire de comparer ces stratégies, en utilisant par exemple des critères objectifs. Nous disposons d'un certain nombre de critères que l'on a sélectionnés pour notre planification, cependant la comparaison selon ces critères est biaisée puisque notre planification est effectuée en fonction de ces mêmes critères. Pour obtenir une validation précise et plus objective, il serait nécessaire de pouvoir comparer les stratégies à l'aide d'outils ou personnes indépendantes de la façon dont l'une ou l'autre des stratégies a été sélectionnée. Il serait alors nécessaire de demander à un ou plusieurs médecins expérimentés indépendants de celui qui a pratiqué l'opération, de noter les stratégies comparativement. Pour effectuer cette validation, il est nécessaire de mettre en place un suivi des opérations de radiofréquence pratiquées localement sur plusieurs mois. Cela implique d'établir un protocole d'acquisition d'images pré-opératoires et per-opératoires de façon à collecter les données indispensables à la planification automatisée et les données concernant la stratégie effectuée. Chaque cas doit alors être reconstruit en 3D dans son intégralité et recalé avec les données disponibles sur la stratégie d'intervention pratiquée. Enfin la notation des stratégies correspondant aux différents cas rassemblés requiert l'organisation d'entretiens avec des spécialistes comprenant des intervenants indépendants des interventions concernées. Cette validation permettrait alors d'évaluer la pertinence de notre outil de planification et d'ajuster directement la définition des contraintes, en fonction des indications des spécialistes.

## Chapitre 4

# Résolution des contraintes de planification

### Sommaire

---

<b>4.1</b>	<b>Approche objet pour l'évaluation des contraintes</b>	<b>87</b>
4.1.1	Evaluation d'un terme	87
4.1.1.1	Processus récursif d'évaluation	87
4.1.1.2	Noeuds à valeurs variables	89
4.1.2	Limitations des évaluations redondantes	89
<b>4.2</b>	<b>Un opérateur spécifique pour le problème de l'évitement d'obstacles</b>	<b>91</b>
4.2.1	Le problème de la contrainte d'évitement d'obstacles	91
4.2.2	Le traitement des obstacles grâce à un algorithme de calcul de visibilité	92
4.2.2.1	Détermination de la visibilité depuis un point	92
4.2.2.2	Extension de l'algorithme pour calculer la visibilité depuis un volume	94
<b>4.3</b>	<b>Le problème d'optimisation sous contraintes à résoudre</b>	<b>95</b>
4.3.1	Réduction des degrés de liberté	95
4.3.1.1	Suppression du paramètre de puissance	95
4.3.1.2	Fonction associant placement et puissance dans le cas de la RFA	96
4.3.1.3	Simulation physique du volume d'effet	96
4.3.2	Séparation du traitement des contraintes strictes et des contraintes souples	97
<b>4.4</b>	<b>Détermination de l'espace des solutions</b>	<b>99</b>
4.4.1	Les contraintes strictes d'affectation	99
4.4.2	Les contraintes strictes restrictives	100
<b>4.5</b>	<b>Optimisation des contraintes souples</b>	<b>103</b>
4.5.1	Problématique générale	103
4.5.2	Echantillonnage de l'espace des solutions	104
4.5.3	Choix des échantillons d'intérêt	105
<b>4.6</b>	<b>Validation de l'heuristique sur un exemple de fonction objectif</b>	<b>107</b>

---

4.6.1	Algorithme quasi-exhaustif pour le calcul de l'optimum global de la fonction étudiée . . . . .	108
4.6.1.1	Élimination du problème des minima locaux en fixant une partie des paramètres . . . . .	108
4.6.1.2	Méthode de calcul du minimum théorique . . . . .	111
4.6.2	Comparaison des résultats fournis par l'heuristique proposée avec le minimum théorique . . . . .	111
<b>4.7</b>	<b>Résolution des contraintes : conclusion et perspectives . . .</b>	<b>112</b>

---



APRÈS avoir présenté le langage formel permettant de décrire les contraintes de planification d'une opération de chirurgie mini-invasive, nous allons maintenant détailler les méthodes développées pour leur résolution. Dans un premier temps, nous présenterons l'approche objet que nous avons choisie pour la résolution des contraintes décrites et en particulier l'évaluation des termes représentant les contraintes. Nous nous intéresserons alors à la sémantique opérationnelle d'un opérateur particulier permettant de résoudre de façon efficace le problème d'évitement d'obstacle, intervenant de façon récurrente dans les problématiques de planification de chirurgie mini-invasive. Puis nous exposerons le problème d'optimisation sous contraintes tel qu'il est formalisé et nous détaillerons comment nous l'avons interprété et simplifié en vue de sa résolution. Nous développerons alors le traitement spécifique des contraintes selon leur nature : la détermination du domaine des solutions grâce à la résolution des contraintes strictes, puis l'optimisation des contraintes souples à l'intérieur de ce domaine. Enfin nous tenterons d'évaluer l'efficacité de notre méthode d'optimisation sur une fonction objectif particulière.

## 4.1 Approche objet pour l'évaluation des contraintes

Les contraintes décrites à l'aide de notre langage doivent être résolues par le solveur de contraintes pour proposer une solution compatible. Traditionnellement, deux approches sont envisageables lorsqu'il s'agit de faire appliquer par la machine des instructions décrites dans un langage formel. La première est de passer par une étape de compilation, la seconde est d'interpréter les instructions pendant l'exécution. L'étape de compilation permet de gagner en efficacité, cependant elle implique une perte de modularité par rapport à une interprétation directe. C'est précisément la modularité qui nous a motivé à considérer les contraintes de planification comme des données à décrire dans un langage spécifique indépendant du solveur. L'interprétation du fichier de contraintes à l'exécution permet de conserver cette caractéristique : l'ajout et la modification de nouvelles contraintes sont immédiats et peuvent être facilités pour l'utilisateur à travers une interface graphique. De plus, si l'on souhaite dans le futur rendre le langage facilement extensible par l'ajout de nouveaux opérateurs, cette modularité est indispensable. L'objectif de la rapidité d'évaluation n'est pas forcément compatible avec la contrainte de modularité, cependant les progrès dans les langages objets et leur compilateur nous permettent d'envisager une approche orientée objet de l'interprétation des contraintes suffisamment efficace pour être applicable en un temps raisonnable.

### 4.1.1 Évaluation d'un terme

#### 4.1.1.1 Processus récursif d'évaluation

Le coeur de la résolution d'une contrainte, quel que soit son type, concerne l'évaluation du terme qui lui est associé avec différentes valeurs des paramètres le cas échéant. La question de l'interprétation et de l'évaluation d'expressions dans un langage formel est classique en informatique. Nous appliquons une approche objet bien connue et documentée

qui consiste à traduire les termes en arbres syntaxiques abstraits et effectuer les évaluations du terme par l'intermédiaire de cette structure de donnée facilement manipulable (design pattern *interpreter* [Deb07]).

L'arbre syntaxique correspondant à un terme a été décrit à la section 3.1.3. Du point de vue objet, les acteurs intervenant dans la construction et l'évaluation de cet arbre sont les suivants :

- la classe *Noeuds* qui correspond à une abstraction regroupant les classes *Noeud Interne* et *Feuille*. Un *Noeud* est associé à la valeur du sous-terme dont il est la racine et doit être capable de s'évaluer.
- la classe *Opérateurs* qui représente également une abstraction de l'ensemble des classes décrivant un opérateur du langage (ex : *plus*, *moins*, *distmin*, *angle*). Un opérateur doit être capable d'appliquer l'opération associée sur des opérandes compatibles quelconques (par exemple dans le cas de l'opérateur *plus*, la fonction de résolution est simplement  $plus(\text{operandes}) = \text{operandes}_1 + \dots + \text{operandes}_n$ ).

Le procédé d'évaluation d'un noeud est le suivant :

- dans le cas d'un noeud feuille, l'évaluation est immédiate et correspond à la donnée stockée.
- dans le cas d'un noeud interne l'approche d'évaluation est compositionnelle : le noeud demande l'évaluation de chacun de ses noeuds fils puis applique l'opérateur associé au noeud sur les résultats récupérés.

Pour évaluer un terme à un instant donné, il suffit alors d'envoyer une requête d'évaluation de la racine de l'arbre syntaxique du terme. La requête d'évaluation est transmise alors dans l'arbre selon un parcours en profondeur suffixé illustré par la figure 4.1.

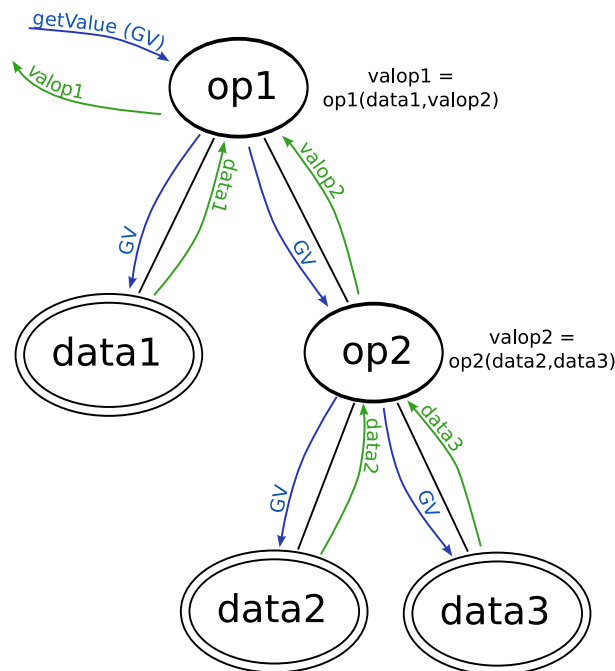


FIGURE 4.1: Evaluation d'un terme

#### 4.1.1.2 Noeuds à valeurs variables

Dans l'expression des termes, on rencontre deux types de noeuds :

- les noeuds dont la valeur est fixe pendant une phase de résolution, elle dépend uniquement de la valeur des données manipulées (nous évoquerons dans la section 5.4 le cas où les valeurs des données sont modifiées par l'utilisateur)
- les noeuds racines d'un sous-arbre qui représente des fonctions de la stratégie étudiée. Leurs valeurs évoluent tout au long de la résolution. Ces noeuds se distinguent des autres par l'introduction d'un opérateur faisant intervenir la stratégie courante dans le sous-arbre associé.

L'évaluation d'un noeud variable se fait donc en 2 étapes. La première consiste à remplacer l'inconnue dans l'arbre syntaxique par la stratégie courante étudiée. La seconde étape correspond à l'évaluation classique du noeud comme s'il s'agissait d'un noeud constant.

#### 4.1.2 Limitations des évaluations redondantes

Le fonctionnement modulaire des termes implique une difficulté à optimiser les calculs redondants et donc un coût plus important en terme de temps d'exécution. Chaque noeud ayant une procédure d'évaluation indépendante des autres, le procédé de résolution ne permet pas de réutiliser des calculs qui ont été nécessaires à la résolution d'un autre noeud de l'arbre. En effet, ne connaissant pas les contraintes a priori, il n'est pas possible de déterminer qu'un calcul effectué pour la résolution de telle opération pourrait être réutilisé dans la suite de l'évaluation. Etant donné que les termes représentant les contraintes doivent pouvoir être évalués un nombre conséquent de fois pour différentes stratégies étudiées, il est nécessaire de proposer des aménagements de la structure de donnée représentant la contrainte et de la méthode d'évaluation. La question de la limitation des évaluations des expressions au strict nécessaire intervient fréquemment dans de nombreux langages. L'approche d'évaluation réduite au minimum est connue sous le nom d'*évaluation paresseuse* et a été largement étudiée par la communauté de programmation fonctionnelle [PJ87][Hud89]. Cette stratégie comporte différentes techniques :

- le report de l'évaluation des expressions au moment où leur valeur est utilisée.
- le partage d'expressions identiques par l'intermédiaire de pointeurs. Cela permet d'alléger la taille des structures de données décrivant les expressions ainsi que d'empêcher l'évaluation multiple de la même expression causée par une structure de donnée redondante. Cela a pour conséquence de transformer les arbres syntaxiques en graphes orientés.
- la mémorisation des résultats dans l'arbre syntaxique en vue d'une éventuelle réévaluation.

Ces techniques ont été en partie mises en oeuvre pour notre solveur de contraintes. L'évaluation des données supplémentaires définies dans le fichier de contraintes n'est effectuée que lorsqu'elle est nécessaire pour la résolution d'une contrainte ou explicitement demandée. Le partage de sous-termes identiques à l'intérieur d'un graphe syntaxique ou entre différents graphes de contraintes est possible grâce à la définition d'une donnée sup-

plémentaire décrivant le sous-terme utilisé à plusieurs reprises. Cependant cette détection de la redondance de termes n'est à l'heure actuelle pas automatique et nécessite un effort lors de la rédaction du fichier de contraintes. La mémorisation des résultats est également utilisée : lorsqu'un noeud interne s'évalue, il stocke localement la valeur qui pourra être directement réutilisée lors de la prochaine évaluation. Dans notre cas il est possible de manipuler des noeuds dont la valeur est constante au cours de la résolution et d'autres dont la valeur dépend de la stratégie courante. La valeur stockée ne sera réutilisée pour l'évaluation suivante uniquement dans le cas d'un noeud à valeur constante.

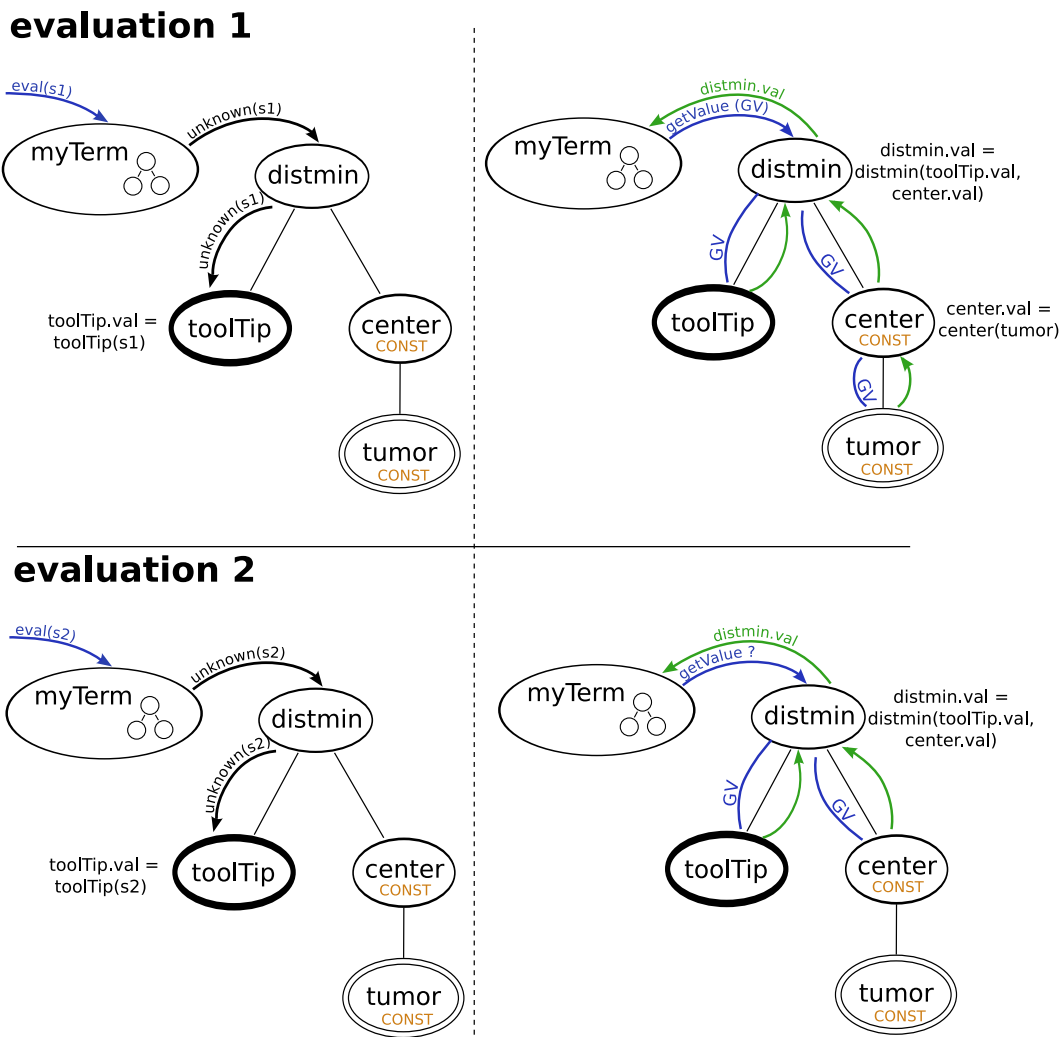


FIGURE 4.2: Exemple de scénario d'évaluations successives d'un même terme : L'évaluation du terme se déroule en 2 étapes : propagation de la valeur de l'inconnue puis évaluation. Les requêtes d'évaluation sont en bleu et les réponses en vert.

Illustrons le fonctionnement de l'évaluation d'un terme et de la mémorisation des résultats intermédiaires grâce à un scénario d'évaluation représenté sur la figure 4.2. Lors de la première requête d'évaluation, le terme reçoit  $s_1$  comme stratégie courante. Il prévient son noeud racine *distmin* qu'à partir de maintenant l'inconnue vaut  $s_1$ , celui-ci répercute l'in-

formation à son unique noeud variable *toolTip*. *toolTip* étant un noeud inconnu il se définit comme valeur courante  $toolTip(s_1)$ . Le terme peut alors demander à sa racine de s'évaluer, puisque toutes les données constituant le terme sont maintenant fixées. La demande d'évaluation se propage dans tout l'arbre, et chacun des noeuds stocke localement la valeur de cette évaluation. Une fois toutes ces opérations effectuées le terme peut renvoyer comme résultat de son évaluation la valeur de son noeud racine. Lors de la deuxième requête d'évaluation avec  $s_2$  comme stratégie courante, la mise-à-jour de l'inconnue se déroule comme la première fois. En ce qui concerne la nouvelle évaluation elle ne se propage pas dans l'arbre en entier : lorsque le noeud *distmin* demande à son fils *center* de s'évaluer, celui-ci renvoie directement la valeur stockée lors de la première évaluation puisque sa valeur est constante, évitant ainsi de recalculer le centre de la tumeur.

## 4.2 Un opérateur spécifique pour le problème de l'évitement d'obstacles

La méthode d'évaluation des termes que nous avons détaillée nous permet de résoudre des contraintes relativement complexes dans un délai raisonnable. Cependant, la résolution de la contrainte d'évitement des organes vitaux nécessite a priori de nombreux calculs qui impliqueraient un temps d'exécution conséquent, amplifié par le fonctionnement modulaire du solveur. Cette problématique est particulièrement fréquente dans la planification d'interventions de chirurgie mini-invasives, la plupart des interventions impliquant des zones critiques à éviter. Nous avons donc développé un opérateur spécifique permettant de répondre à ce problème de façon précise tout en optimisant la rapidité de la résolution.

### 4.2.1 Le problème de la contrainte d'évitement d'obstacles

La contrainte d'évitement d'obstacle pourrait être exprimée comme une autre contrainte stricte grâce au terme suivant, avec une contrainte par obstacle considéré :

$$not(cross(toolTrajectory, obstacle_i))$$

Cette façon de définir la contrainte d'évitement des obstacles s'accompagne de plusieurs inconvénients. Tout d'abord la résolution de cette contrainte ne sera pas effectuée avec une efficacité optimale. D'une part, selon la façon dont est résolu l'opérateur *cross*, celui-ci peut impliquer le calcul d'une intersection qui est relativement coûteux, surtout s'il est répété à de nombreuses reprises. Or il ne s'agit pas ici de calculer l'intersection avec un volume, mais de déterminer s'il y a une intersection ou non. Ce calcul des collisions peut être effectué de façon rapide grâce à la carte graphique en considérant que le fait qu'il n'y ait pas d'obstacle sur un placement d'outil est équivalent au fait de ne pas voir d'obstacle lorsque l'on se place au niveau de la pointe de l'outil et que l'on regarde dans la direction du point d'insertion de l'outil (voir article [LCN99]). Cependant, même en déterminant les collisions de cette façon, le fait de séparer le traitement de chaque obstacle entraîne des

opérations redondantes et donc inutilement coûteuses. Nous nous sommes donc inspirés de cette méthode considérant les obstacles du point de vue de leur visibilité depuis la cible pour proposer un opérateur particulier nommé *visible* permettant à partir de deux formes de déterminer quelles sont les parties complètement visibles de la première forme si l'on se place dans la deuxième. Cet opérateur est accompagné de l'ajout d'un attribut spécifique aux formes constituant la scène indiquant si la forme est un obstacle à considérer (attribut *obstacle*, cf section 3.4.2.3). Par défaut toute forme appartenant à la scène est un obstacle. Cette caractéristique peut être modifiée grâce à la partie `< data_definition >` dans le fichier définissant les contraintes.

## 4.2.2 Le traitement des obstacles grâce à un algorithme de calcul de visibilité

Nous allons maintenant détailler comment fonctionne l'opérateur *visible* :  $Forme * Forme \rightarrow Forme$  qui détermine à partir de deux formes les parties complètement visibles de la première forme si on se place à un endroit quelconque de la deuxième (voir figure 4.3).

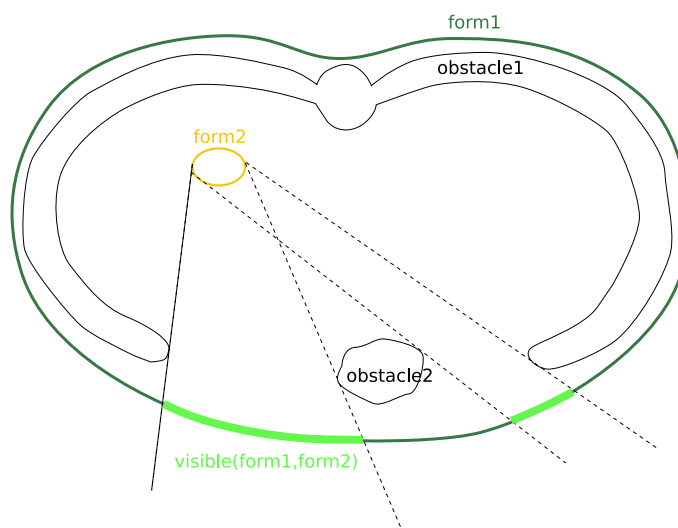


FIGURE 4.3: Opérateur visible

### 4.2.2.1 Détermination de la visibilité depuis un point

Considérons tout d'abord la visibilité depuis un unique point dans l'espace. On cherche à déterminer les parties visibles de la forme considérée depuis ce point. Nous avons déjà évoqué qu'un objet de type *Forme* peut être soit considéré sous forme de surface (maillage) soit sous forme de volume (ensemble de voxels). Dans le cas de cet opérateur, la donnée manipulée est le maillage de la forme dont on veut déterminer la visibilité. Le principe général de l'algorithme pour déterminer le sous maillage visible depuis un point est similaire

à l'algorithme 4.1 qui calcule les sous-parties d'un maillage respectant une contrainte. Il s'agit de considérer chaque triangle du maillage, le garder s'il est complètement visible, l'éliminer s'il est complètement caché, le subdiviser et réévaluer ses sous-triangles s'il n'est que partiellement visible.

Une vue d'une scène 3D (voir figure 4.4(a)) est principalement définie par la position de la caméra virtuelle, une direction de visualisation ainsi que des limites de visibilité en termes d'angles et de distance. Si l'on considère un cube imaginaire autour de la position de la caméra, il est possible de visualiser tout l'espace grâce à 6 vues, chacune des vues correspondant à la projection de la scène sur une des faces du cube. Pour déterminer la visibilité d'un triangle il faut donc examiner chacune de ces 6 vues. Dans chaque vue on compare alors le nombre de pixels<sup>1</sup> que représente le triangle (figure 4.4(b)) avec le nombre de pixels représentant le triangle en ignorant les obstacles (figure 4.4(c)). Il y a alors trois cas possibles :

- Le nombre de pixels représentant le triangle est le même, que l'on tiennent compte ou non des obstacles. Le triangle est alors complètement visible.
- Le nombre de pixels représentant le triangle est différent si l'on tient compte des obstacles, mais est différent de zéro. Cela signifie que le triangle est partiellement caché par un obstacle (cas représenté sur la figure 4.4).
- Le nombre de pixels représentant le triangle est différent si l'on tient compte des obstacles et est égal à zéro. Le triangle est donc complètement caché.

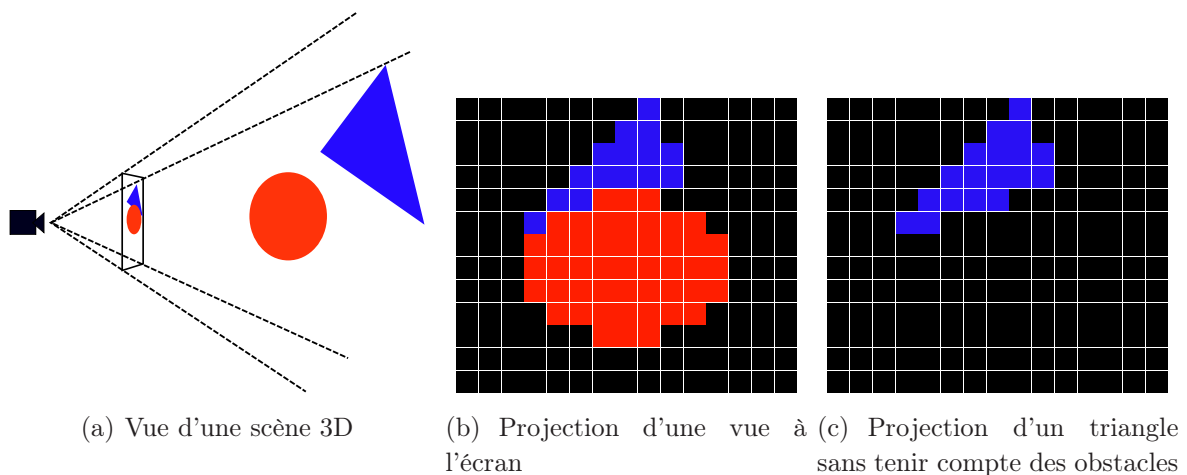


FIGURE 4.4: **Détermination de la visibilité d'un triangle dans une vue** : le triangle en bleu est représenté à l'écran par un nombre différent selon que l'obstacle rouge est affiché ou non, il est donc partiellement visible dans la vue considérée

1. l'extension opengl ARB\_OCCLUSION\_QUERY permet de connaître le nombre de pixels correspondant à l'affichage d'une primitive opengl

### 4.2.2.2 Extension de l'algorithme pour calculer la visibilité depuis un volume

Nous souhaitons déterminer la visibilité de la forme considérée non pas depuis un point précis, mais depuis un endroit quelconque d'une forme. Il suffit d'utiliser la discrétisation de la forme représentée comme un ensemble de voxels et d'appliquer l'algorithme que l'on vient de décrire en considérant successivement chaque voxel comme placement de la caméra. Cependant pour optimiser le temps de calcul correspondant, on ne considère que les voxels du bord du volume, et pour ces voxels que certaines vues dirigées vers l'extérieur du volume. En effet considérer la visibilité dans un axe depuis un point à l'intérieur du volume est équivalent à se placer au bord de ce volume en suivant cet axe. La figure 4.5 schématise les calculs de visibilité à effectuer sur un cas simple en 2 dimensions.

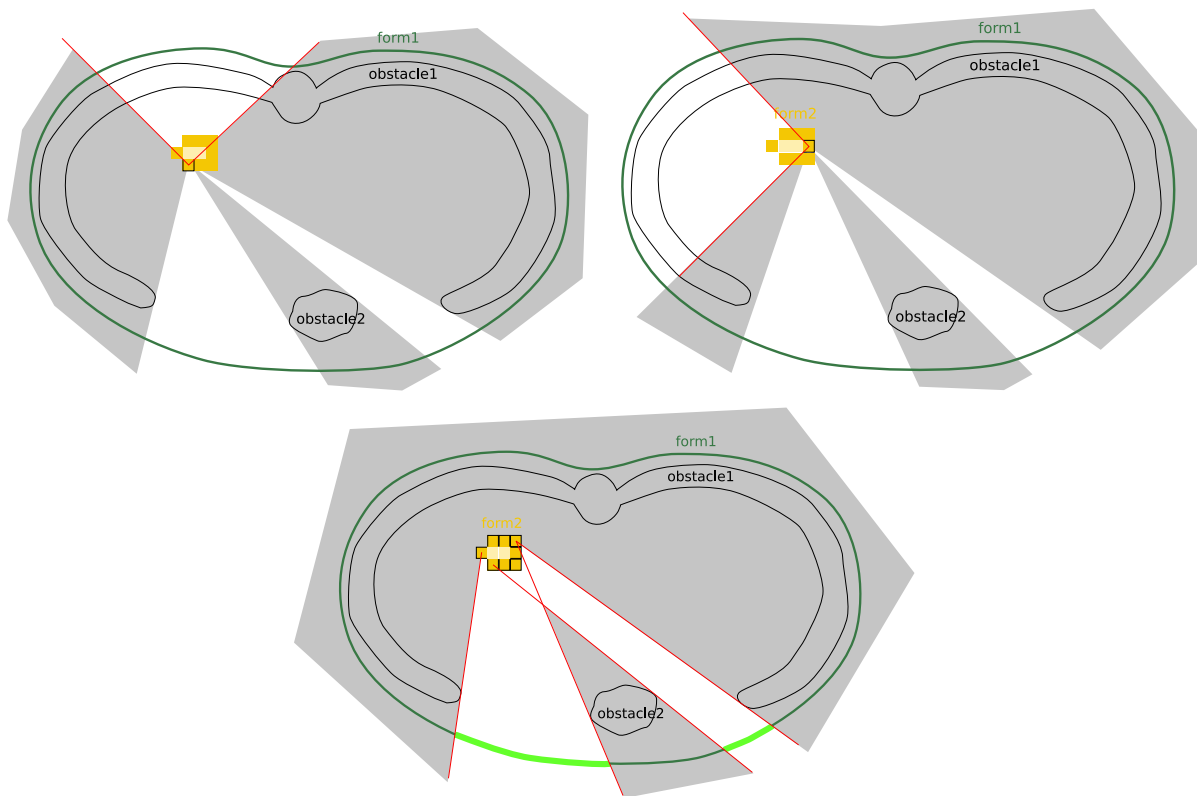


FIGURE 4.5: **Détermination de la visibilité depuis un volume** : les voxels considérés sont en jaune, et pour chacun d'eux les vues évaluées sont en gras. Les zones grisées représentent les parties de la vue cachées par les obstacles. Les deux premières figures représentent le résultat correspondant au calcul de la visibilité à partir de deux voxels particuliers du volume. La troisième figure représente le résultat final une fois tous les calculs effectués dans toutes les vues, la partie visible de *form1* depuis *form2* est alors la zone verte en gras.



## 4.3 Le problème d'optimisation sous contraintes à résoudre

Nous allons ici exposer le problème tel qu'il est défini en théorie par les contraintes, puis nous verrons comment ce problème est modifié pour pouvoir faciliter sa résolution. Le problème auquel on s'intéresse est le suivant :

$$\begin{aligned} & \text{minimiser } \textit{soft\_constraint}_{agr}(S) \\ & \text{avec } \textit{strict\_constraint}_i(S), \forall i \in [0, n] \end{aligned}$$

$S$  est une stratégie et appartient donc à  $R^6$ . La fonction  $\textit{soft\_constraint}_{agr} : Strategy \rightarrow float$  est la contrainte souple d'agrégation. Les fonctions  $\textit{strict\_constraint}_i : Strategy \rightarrow bool$  sont les contraintes strictes à respecter,  $n$  le nombre de ces contraintes.

La résolution d'un problème d'optimisation dans un domaine continu à six dimensions peut être très coûteuse en termes de temps de calcul, nous avons donc cherché à simplifier ce problème théorique de façon à ce que sa résolution soit envisageable dans un temps raisonnable.

### 4.3.1 Réduction des degrés de liberté

La première étape de simplification consiste à diminuer les degrés de liberté de l'espace de recherche. Rappelons que la *stratégie* recherchée est un élément de  $R^6$  : 3 paramètres correspondent au placement de la pointe de l'outil, 2 paramètres correspondent aux angles représentant l'inclinaison de l'aiguille en coordonnées polaires, et un dernier paramètre correspond à la puissance de l'effet du traitement. Nous avons diminué la taille de l'espace de recherche en fixant ce dernier paramètre en fonction des autres.

#### 4.3.1.1 Suppression du paramètre de puissance

Considérons un *placement* d'outil  $P$  quelconque, c'est-à-dire un élément de  $R^5$  correspondant à une position de la pointe et deux angles pour l'axe. En supposant qu'il n'y ait pas de limite de puissance pour l'outil, et donc que la taille de la zone d'effet résultante soit illimitée, il existe pour chaque placement d'outil une infinité de valeurs possibles pour la puissance pour former une stratégie à étudier. En réalité, toutes les puissances possibles pour une stratégie ne sont pas intéressantes à étudier. Quel que soit l'énoncé des contraintes, on sait qu'un volume à traiter a été défini. On suppose que la fonction qui lie la puissance de l'outil et le volume de la zone d'effet est strictement croissante. On sait donc que pour un placement d'outil donné, ne nous intéressent que les valeurs de puissance qui permettent d'englober tout le volume à traiter dans la zone d'effet. D'autre part, il n'est pas nécessaire que la puissance choisie permette d'englober un volume beaucoup plus grand que la zone d'ablation. Dans le cas de la radiofréquence, le volume des tissus nécrosés devra contenir le moins possible de tissus sains. Même si, en théorie, une certaine puissance permet d'inclure le patient entier dans la zone d'effet de l'outil, il est évident que

ce n'est pas souhaitable et qu'on se contentera de la puissance nécessaire et suffisante pour englober la zone d'ablation.

Il existe donc une fonction qui à partir d'un *placement* d'outil et d'un volume à traiter permet de déduire la *stratégie* correspondant à une valeur de puissance nécessaire et suffisante. Cette fonction associe au placement  $P : (x, y, z, \alpha, \beta)$  la stratégie  $S : (x, y, z, \alpha, \beta, pow)$  telle que :

- $cover(S, treatmentVolume)$  et
- $\forall S' : (x, y, z, \alpha, \beta, pow')$  tel que  $pow' < pow, -cover(S', treatmentVolume)$ .

#### 4.3.1.2 Fonction associant placement et puissance dans le cas de la RFA

Pour l'application de ces travaux, nous nous sommes intéressés à l'ablation par radiofréquence percutanée. Nous avons évoqué précédemment (voir section 1.3.1) que nous avons choisi de modéliser le volume de nécrose<sup>2</sup> sous la forme d'un sphéroïde dont le rapport entre les axes est fixe. Pour chaque placement d'aiguille, l'objectif est de trouver le sphéroïde (ou la puissance associée) de taille minimale englobant le volume à traiter.

On se place dans le repère  $(O, \vec{x}, \vec{y}, \vec{z})$ ,  $O$  étant la position de la pointe de l'aiguille,  $\vec{x}$  le vecteur directeur de l'aiguille et  $\vec{y}$  et  $\vec{z}$  deux vecteurs quelconques tels que le repère soit orthonormé. L'équation du sphéroïde représentant la zone de nécrose est la suivante :

$$\frac{x^2}{r^2} + \frac{y^2}{k^2 \cdot r^2} + \frac{z^2}{k^2 \cdot r^2} = 1$$

$r$  représente la longueur du demi-axe principal, et  $k$  le rapport entre le demi-axe principal et les axes secondaires. Pour déterminer la valeur du demi-grand axe du sphéroïde minimal permettant d'englober en totalité la zone d'ablation, on procède de manière itérative sur les données représentant la zone d'ablation. On teste si chacun des points de son bord est à l'intérieur du sphéroïde courant, si ce n'est pas le cas on actualise le rayon du sphéroïde de façon à ce qu'il passe par le point concerné. Le sphéroïde résultant est alors suffisamment grand pour englober la zone d'ablation, mais diminuer son rayon impliquerait qu'il y ait au moins un point de la zone d'ablation non considéré.

#### 4.3.1.3 Simulation physique du volume d'effet

La modélisation du volume nécrosé par radiofréquence sous la forme d'un sphéroïde n'est qu'une approximation rapide permettant d'effectuer de nombreuses tentatives pour le processus d'optimisation. Plusieurs travaux de recherche ont porté sur la simulation plus précise de cette nécrose grâce à des modèles physiques. Nous avons écarté ce type d'approche pour des raisons pratiques, une modélisation plus élaborée étant à l'heure actuelle trop gourmande en temps de calculs pour pouvoir être utilisée à de très nombreuses reprises. Cependant, en supposant que par la suite ce genre de simulation soit plus efficace, on pourrait, de la même manière qu'on l'a fait avec notre sphéroïde, associer un placement

---

2. le volume de nécrose correspond au volume d'effet de l'outil dans le cas d'une aiguille de radiofréquence

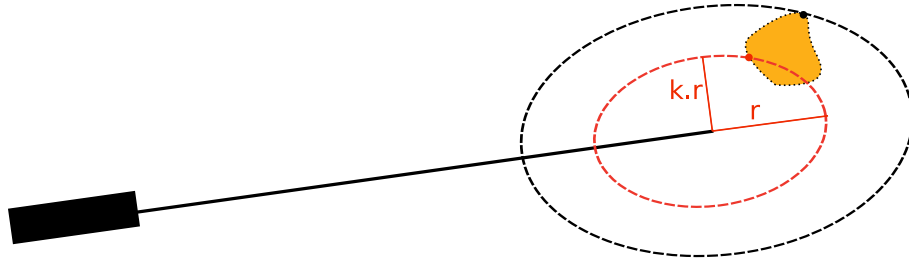


FIGURE 4.6: **Volume de nécrose correspondant à un placement d'aiguille** : le sphéroïde de base de l'outil dont le rapport entre les axe est fixé est agrandi de façon à englober la zone d'ablation en totalité

avec une stratégie en augmentant la puissance jusqu'à ce que le volume de nécrose puisse inclure le volume à traiter. L'association entre un placement et une stratégie est aussi valable si l'on considère d'autres types d'outils dont le volume d'effet est croissant en fonction d'un paramètre de puissance.

### 4.3.2 Séparation du traitement des contraintes strictes et des contraintes souples

Le problème tel que nous l'avons exprimé au début de cette section sous-entend un traitement simultané des contraintes souples et strictes, c'est-à-dire que l'on cherche à minimiser la fonction objectif représentant les contraintes souples tout en vérifiant que nos contraintes strictes sont respectées. L'approche de résolution accompagnant cette formulation du problème implique de considérer toutes les valeurs possibles pour les 5 paramètres représentant un placement d'outil, et pour chaque tentative étudiée durant le processus d'optimisation, vérifier préalablement que les contraintes strictes sont respectées. Les vérifications des contraintes strictes nécessaires tout au long du processus d'optimisation sont coûteuses et l'alourdissent. De plus la visibilité globale des solutions du problème n'est pas naturelle à travers cette approche. Nous avons donc décidé de traiter en premier lieu les contraintes strictes de façon à fournir un sous-ensemble de  $R^5$  qui ne comporte que des solutions du problème. Les contraintes souples seront alors traitées dans un second temps grâce à une optimisation réduite à l'espace de recherche défini par les contraintes strictes. L'heuristique développée pour cette phase d'optimisation a alors la possibilité d'exploiter la matérialisation de l'espace des solutions pour effectuer efficacement la recherche à l'intérieur de celui-ci.

Les ensembles de délimitation *insertionZone*, *targetVolume*, *treatmentVolume* déterminent l'espace des stratégies solutions (voir section 3.3.2). *treatmentVolume* nous permet d'associer à chaque placement (élément de  $R^5$ ) une stratégie (élément de  $R^6$  : placement + paramètre de puissance de l'outil) de façon à ce que la recherche puisse être réduite en dimension 5. Cet ensemble n'est pas lié à la séparation du traitement de nos deux types de contraintes. *targetVolume* est l'espace résultant de la projection de l'espace des solutions sur les trois paramètres représentant la pointe de l'outil. *insertionZone* résulte de

la projection des 5 paramètres représentant un placement sur une surface prédéterminée. Notre approche nous conduit donc à calculer explicitement ces deux ensembles, de façon à ce qu'en parcourant l'espace des placements possibles qu'ils définissent, le respect des contraintes strictes soit assuré.

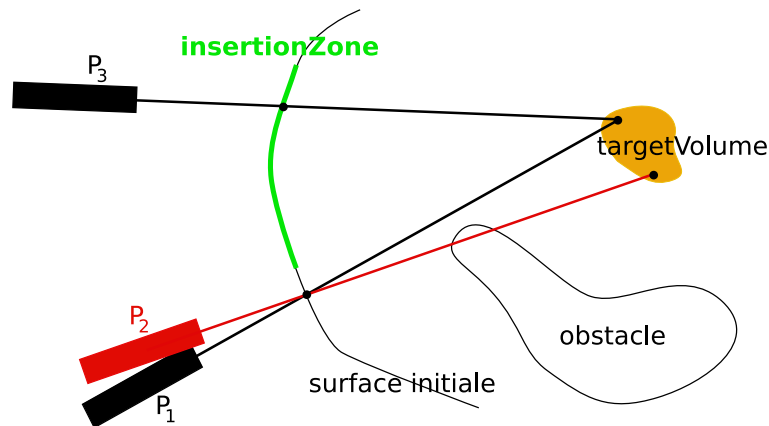


FIGURE 4.7: **Limite de la représentation de l'espace des solutions** : si l'on considère comme contrainte stricte le fait de ne pas traverser les obstacles, le placement  $P_1$  est une solution mais n'est pas considéré comme tel puisque le point d'insertion correspondant peut mener à un placement interdit comme  $P_2$

La contrepartie de ce choix est qu'il n'est pas possible de représenter tous les sous-ensembles de  $R^6$  sous cette forme. La figure 4.7 illustre ce problème. Soient 3 placements d'outils distincts  $P_1$ ,  $P_2$  et  $P_3$ . Les placements  $P_1$  et  $P_2$  partagent le même point d'insertion, alors que  $P_1$  et  $P_3$  partagent la même position de la pointe. Si l'on considère comme contrainte stricte uniquement le fait que l'aiguille ne doit pas traverser l'obstacle, alors les placements  $P_1$  et  $P_3$  représentés sur la figure sont des solutions alors que  $P_2$  ne l'est pas. Avec notre représentation de l'espace des solutions, le fait que  $P_1$  et  $P_3$  appartiennent à cet espace implique que les points d'insertion de  $P_1$  et  $P_3$  appartiennent à la zone d'insertion et que la position de la pointe de  $P_1$  et  $P_3$  appartiennent au volume cible. Cela impliquerait cependant que le placement  $P_2$  soit considéré lui aussi comme une solution puisque son point d'insertion (le même que  $P_1$ ) appartient à la zone d'insertion et que la position de sa pointe (la même que  $P_3$ ) appartient au volume cible. Pourtant le placement  $P_2$  implique de traverser l'obstacle, il ne doit pas être considéré comme une solution. Nous avons donc choisi de considérer la zone d'insertion de la manière la plus restrictive possible. Tout placement d'outil inséré dans la zone d'insertion doit pouvoir atteindre n'importe quel point du volume cible tout en respectant les contraintes. Si ce n'est pas le cas, ce point n'est pas considéré comme faisant partie de la zone d'insertion, et donc tous les placements d'outil correspondants sont éliminés de l'espace des solutions. Si l'on se ramène à notre exemple  $P_1$ , bien qu'étant une solution du problème du point de vue de la contrainte, n'est pas considéré comme telle puisque le point d'insertion auquel il correspond ne garantit pas toujours un placement acceptable ( $P_2$  étant un contre-exemple tout trouvé). Notre représentation de l'espace des solutions nécessite donc d'ignorer une partie des solutions potentielles afin de

garantir le respect des contraintes strictes dans cet espace. Cependant cette restriction n'est pas réellement pénalisante dans la mesure où les solutions ainsi écartées sont des solutions limites. Ces solutions éliminées sont en réalité peu intéressantes : un léger décalage de placement lors de l'application de la planification dans la salle d'opération impliquerait une violation d'une des contraintes strictes.

## 4.4 Détermination de l'espace des solutions

La résolution des contraintes strictes revient donc à déterminer les trois ensembles *insertionZone*, *targetVolume* et *treatmentVolume* de façon à ce que le respect des contraintes strictes soit assuré lorsqu'on se déplace à l'intérieur du sous-espace des stratégies de  $R^6$  que ceux-ci définissent.

Lorsque nous avons présenté les contraintes strictes dans le chapitre traitant de la formalisation des contraintes, nous avons distingué deux types de contraintes strictes :

- les contraintes strictes d'affectation qui donnent une valeur initiale à l'un de ces trois éléments.
- les contraintes strictes restrictives qui réduisent l'espace des solutions.

### 4.4.1 Les contraintes strictes d'affectation

Les termes correspondant aux contraintes d'affectation contiennent l'un des opérateurs *cross*, *target* et *cover*. La résolution des opérateurs *cross*, *target* et *cover* implique simplement d'affecter une valeur respectivement à la zone d'insertion, le volume cible et le volume à traiter. La résolution d'une telle contrainte consiste alors en l'évaluation du sous-arbre correspondant au paramètre de l'opérateur. Cette valeur est alors affectée à la projection de l'espace des solutions qui nous intéresse. Prenons l'exemple d'une contrainte représentée par le terme suivant :

$$cover(enlarge(tumor, 10)) \tag{4.1}$$

La résolution de cette contrainte implique la résolution du sous-terme *enlarge(tumor, 10)* par le procédé classique de résolution de termes que nous avons décrit au début du chapitre. La forme issue de cette résolution (la tumeur élargie de 10 mm) est alors affectée au volume à traiter. Les stratégies acceptables sont donc uniquement celles dont le volume d'effet couvre la tumeur accompagnée d'une marge de sécurité de 10 mm.

Pour que les contraintes strictes puissent être résolues, il est nécessaire qu'une contrainte initiale soit définie pour chaque élément de l'espace des solutions. Cependant on peut tout à fait imaginer qu'il y ait plusieurs contraintes d'initialisation du même type. Par exemple prenons la contrainte définie par le terme 4.1. Celle-ci peut être décomposée en deux contraintes initiales représentées par les termes :

- *cover(tumor)*
- *cover(enlarge(treatmentVolume))*

Rappelons que l'opérateur *cover* produit une donnée nommée *treatmentVolume* actualisée à chaque utilisation. Le volume à traiter correspond à la forme de la tumeur après

**Algorithme 4.1** Décimation de la zone d'insertion

---

```

1:  $L = \text{triangles zone d'insertion}$  {Liste des triangles à évaluer}
2:  $L' = \emptyset$  {Liste des triangles à conserver}
3: pour tout triangle  $ABC$  de  $L$  faire
4:   si  $\text{strict\_constraint}_i(A)$  et  $\text{strict\_constraint}_i(B)$  et  $\text{strict\_constraint}_i(C)$ 
   alors {respect total de la contrainte}
5:    $L' \leftarrow ABC$ 
6:   sinon si  $\neg \text{strict\_constraint}_i(A)$  et  $\neg \text{strict\_constraint}_i(B)$  et  $\neg \text{strict\_constraint}_i(C)$ 
   alors {violation complète de la contrainte}
7:   ignore  $ABC$ 
8:   sinon {respect partiel de la contrainte}
9:    $I_{AB} = \text{milieu}(A, B)$ ,  $I_{AC} = \text{milieu}(A, C)$ ,  $I_{BC} = \text{milieu}(B, C)$ 
10:   $L \leftarrow [AI_{AB}I_{AC}; I_{AB}BI_{BC}; I_{AC}I_{BC}C; I_{AB}I_{BC}I_{AC}]$ 
11: fin si
12: fin pour
13:  $\text{insertionZone} = L'$ 

```

---

l'interprétation et la résolution de la première contrainte stricte. Cette forme est élargie pendant la résolution de la deuxième contrainte et c'est cette nouvelle forme qui devient le nouveau volume à traiter (qui est définitif dans le cas où il n'y a pas d'autre contrainte stricte affectant ce volume par la suite).

#### 4.4.2 Les contraintes strictes restrictives

Les contraintes strictes restrictives agissent sur la zone d'insertion. Une fois le volume cible et le volume à traiter définis, il se pose la question suivante : où peut-on insérer l'outil à travers la surface d'insertion initiale (généralement la peau) de façon à ce que, lorsqu'on le place dans le volume cible et que l'on applique une puissance nécessaire et suffisante pour couvrir le volume à traiter, le respect des contraintes strictes soit assuré ?

Pour déterminer la zone d'insertion, on procède par décimation successive de la surface initiale et par élimination suivant les différentes contraintes. La donnée initiale est un maillage triangulaire maillé plus ou moins finement. Il s'agit alors de déterminer, pour chaque triangle de ce maillage initial, si tous les points d'insertion impliquent le respect des contraintes, ou si la contrainte est respectée partiellement dans le triangle ou alors pas du tout (voir l'algorithme 4.1).

On peut remarquer dans la formulation que l'algorithme sous-entend que l'on peut évaluer une contrainte stricte pour un point d'insertion. Or pour un même point d'insertion une infinité de placements sont possibles selon la position de la pointe de l'outil. Nous avons décidé de restreindre le placement de la pointe de l'outil au volume cible (défini par les contraintes). Si l'on considère ce volume de façon discrète, un nombre fini de positions de la pointe de l'outil sont envisageables. Pour déterminer si un point d'insertion correspond au respect d'une contrainte, on va tester la contrainte sur tous les placements correspondant

aux différentes positions d'outil possibles. Si tous les tests sont positifs, l'insertion de l'outil à partir de ce point et visant le volume cible implique le respect de la contrainte.

Remarquons également que l'algorithme ajoute de nouveaux triangles issus de la subdivision du triangle courant dans la liste des triangles à traiter, dans le cas du respect partiel de la contrainte. On doit préciser une condition d'arrêt pour laquelle la subdivision n'est pas effectuée même si le respect de la contrainte est partiel. Dans notre cas, nous avons choisi de préciser un niveau de subdivision maximum et nous considérons qu'en-dessous d'une taille limite le triangle n'est pas suffisamment représentatif. Si la subdivision n'est pas autorisée à cause de la condition d'arrêt, le triangle est considéré comme violant la contrainte dans son intégralité et est donc éliminé. L'application de notre algorithme de décimation de la zone d'insertion est illustrée sur la figure 4.8. Sans l'étape subdivision, la zone d'insertion résultante est fortement dépendante du maillage initial. Avec 1 niveau de subdivision, les différentes zones sont globalement bien représentées mais manquent de précision. 3 niveaux de subdivision suffisent pour obtenir une zone d'insertion précise avec un maillage initial relativement grossier. Ce résultat est illustré par la figure 4.9 qui représente la zone d'insertion obtenue pour 0, 1 et 3 niveaux de subdivision autorisés, en considérant comme unique contrainte stricte l'évitement des organes.

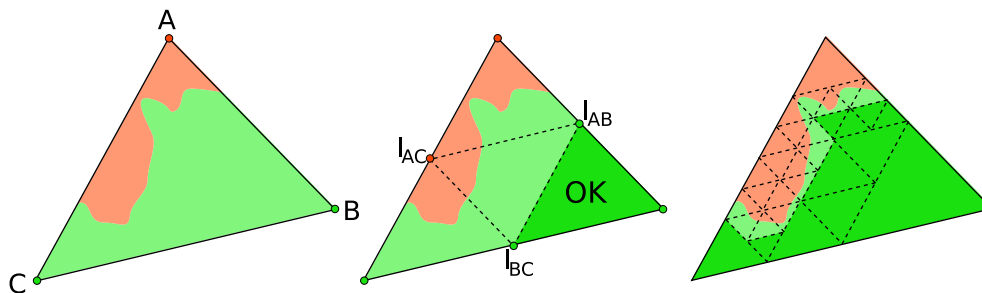


FIGURE 4.8: **Subdivision des triangles pour la détermination de la zone d'insertion :** Les zones colorées respectivement en vert et rouge dans le triangle  $ABC$  représentent le respect / la violation de la contrainte observée. Le triangle  $ABC$  respectant partiellement cette contrainte est subdivisé en 4 sous-triangles qui sont également étudiés vis-à-vis de la contrainte. Les triangles complètement en vert sont conservés dans la zone d'insertion. A un certain niveau de subdivision maximum les triangles partiellement rouges sont retirés sans subdivision supplémentaire.

Notons enfin que l'algorithme considère que la contrainte est respectée dans tout le triangle représentant une partie de la zone d'insertion uniquement à partir des tests portant sur les trois extrémités du triangle. Ces trois tests ne sont évidemment pas suffisants pour garantir le respect de la contrainte dans le triangle. Cependant nous travaillons dans un domaine continu, sur des fonctions dont nous ne connaissons pas a priori les propriétés. Il n'est pas possible de garantir en un nombre fini de tests, quelle que soit la fonction étudiée, que la contrainte est respectée pour tous les placements d'outil traversant une surface. Notre échantillonnage sur les points du maillage représentant la zone d'insertion est suffisant pour la plupart des contraintes que nous avons étudiées dans le cadre de notre étude sur la technique de radiofréquence percutanée. Par exemple dans le cas de la contrainte qui



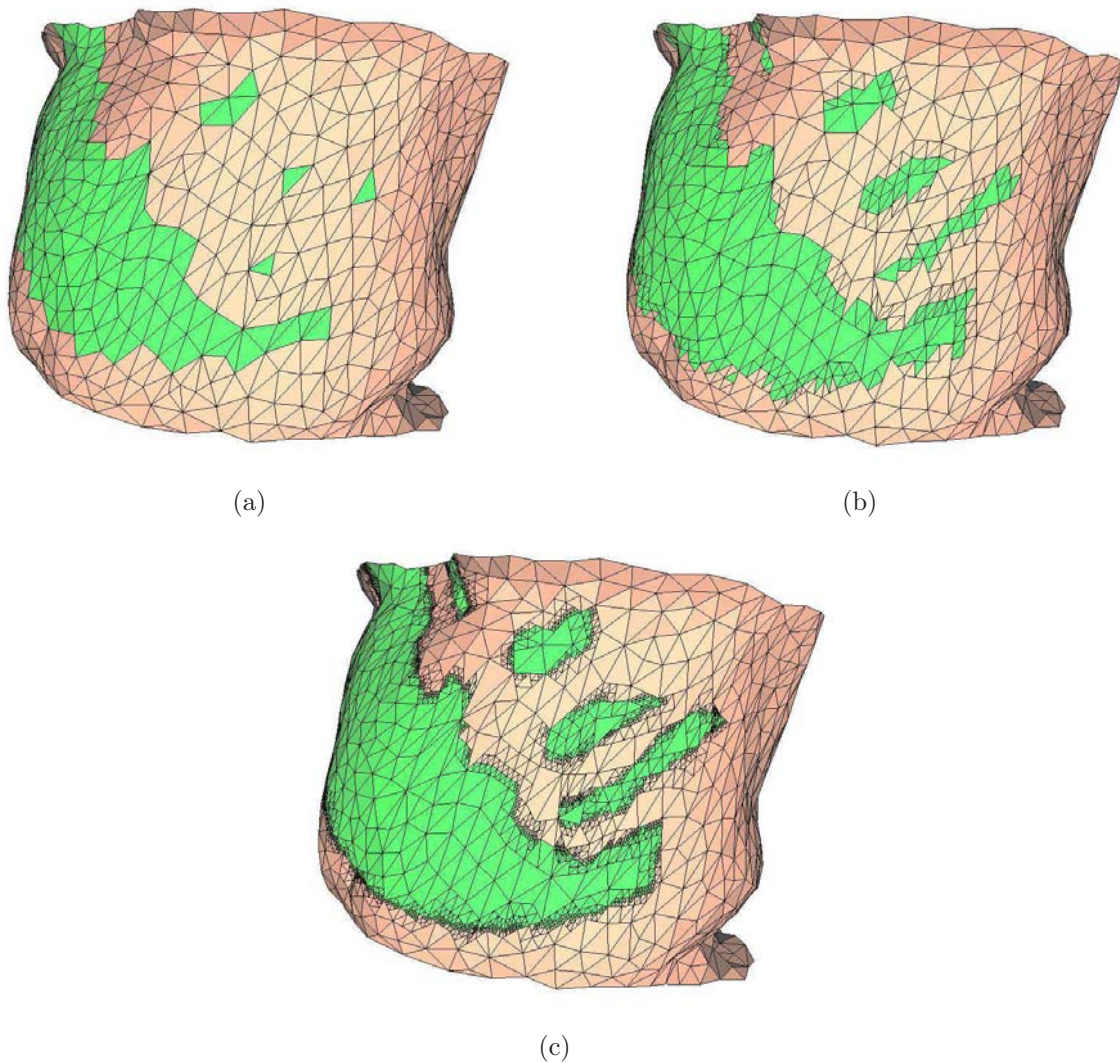


FIGURE 4.9: **Zone d'insertion obtenue avec différents niveaux de subdivision autorisés :**  
a) pas de subdivision b) 1 niveau de subdivision c) 3 niveaux de subdivision

impose que la profondeur du placement soit inférieure à la taille de l'aiguille, l'estimation de la distance à la cible de chaque extrémité du triangle est suffisante pour évaluer le triangle d'insertion dans son intégralité. Par contre en ce qui concerne la contrainte d'évitement des organes vitaux, cette approximation n'est pas suffisante. En effet elle pourrait dans certains cas faire ignorer des obstacles suffisamment fins pour ne pas être considérés par les placements extrêmes du triangle d'insertion (par exemple des gros vaisseaux sanguins). On pourrait envisager d'échantillonner les tests de façon suffisamment fine pour être sûr qu'aucun obstacle ne puisse échapper aux tests. Cependant cette précaution engendrerait un surcoût en terme de calculs conséquent et la plupart du temps inutile puisque toutes les contraintes subissent le même procédé de résolution, qu'elles impliquent ou non l'utilisation d'un échantillonnage fin. Cette constatation a motivé le développement de l'opérateur *visible* (cf. paragraphe 4.2) qui est efficace pour traiter cette contrainte d'évitement d'obs-



tacle, récurrente dans les problématiques de planification de chirurgie mini-invasive .

## 4.5 Optimisation des contraintes souples

### 4.5.1 Problématique générale

La résolution des contraintes strictes aboutit au calcul d'un espace de solution matérialisé sous la forme d'une zone d'insertion et d'un volume cible. Cet espace représente une infinité de placements dans  $R^5$  envisageables, puisque nous travaillons dans un domaine continu. Il s'agit alors de choisir le placement le plus judicieux au regard des contraintes souples. Nous avons déjà évoqué au chapitre 2, la problématique générale qui se pose dans le cas d'un problème d'optimisation à plusieurs objectifs. Il n'est pas possible de proposer une solution optimale vis-à-vis de toutes les contraintes, la solution proposée est un compromis. Nous avons choisi l'approche agrégative qui transforme un problème d'optimisation multi-objectif en problème d'optimisation classique à une seule fonction objectif. La combinaison des différentes contraintes souples pour former la fonction objectif agrégative est définie par la formalisation des contraintes. Il s'agit alors de résoudre un problème d'optimisation classique dans un domaine borné continu, sur une fonction quelconque dont on ne connaît pas le gradient et pouvant comporter des minima locaux.

Nous avons sélectionné pour notre problème un algorithme d'optimisation locale : l'algorithme de Nelder-Mead (voir section 2.2.2.1). C'est un algorithme qui présente l'avantage de converger rapidement et ne nécessite aucun calcul de gradient. Il peut donc s'appliquer à n'importe quelle fonction. Cette propriété est intéressante dans notre cas puisque nous ne savons pas a priori quelle est la fonction que nous cherchons à optimiser. Cependant cette méthode est particulièrement sensible au problème des minima locaux. Prenons l'exemple de la contrainte souple agrégative représentée en dimension 1 sur la figure 4.10a. Sans tenir compte du domaine des solutions, la fonction objectif a un minimum global  $M$  et un minimum local  $A$ . Si l'on applique une méthode d'optimisation sur cette fonction, elle finira probablement par converger soit vers le minimum local  $A$ , soit vers le minimum global  $M$ . Dans ce dernier cas, l'optimisation aurait été une réussite si le point  $M$  correspondait à une variable appartenant au domaine des solutions, or ce n'est pas le cas.

Le premier problème qui se pose est donc le suivant : comment restreindre la recherche dans  $R^5$  à notre domaine des solutions défini par les contraintes strictes ? La solution que nous proposons est de ne pas la restreindre, mais de la bloquer artificiellement en appliquant un malus artificiel et rébarbatif à la fonction objectif de façon à ce que les placements non solution soient naturellement évités par le processus d'optimisation. La fonction objectif modifiée (figure 4.10b) est donc la suivante :

$$obj(x) = \begin{cases} soft\_constraint_{agr}(x) & \text{si } \forall i, strict\_constraint_i(x) \\ max & \text{sinon} \end{cases}$$

avec  $\forall x, max > soft\_constraint_{agr}(x)$

Le problème des minima locaux est inhérent à la plupart des problèmes d'optimisation. Ce problème est aggravé par notre modification de la fonction objectif qui introduit

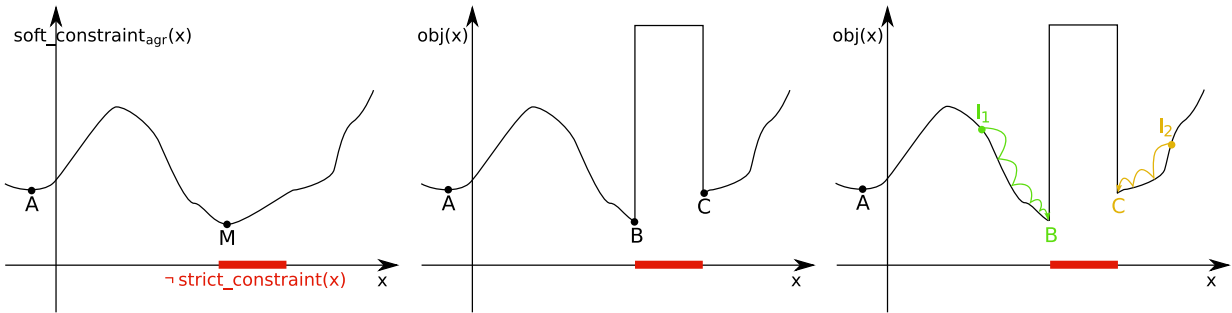


FIGURE 4.10: **Optimisation locale sur une fonction objectif modifiée de façon à éviter les parties interdites de l'espace des paramètres**

beaucoup de minima locaux artificiels (point  $B$  et  $C$  sur la figure 4.10). Dans le cas de méthodes d'optimisation locale, particulièrement sensibles à ce problème, la convergence du processus d'optimisation dépend fortement de l'endroit de l'espace des paramètres où est initialisée la recherche. Sur la figure 4.10c, une recherche locale initialisée à partir du point  $I_1$  aboutira au minimum global  $B$ . Un processus d'optimisation initialisé en  $I_2$  sera bloqué par la barrière créée pour éviter les valeurs non solution et proposera donc le minimum local  $C$ . Il s'agit alors de choisir judicieusement le ou les points d'initialisation du processus d'optimisation de façon à ce que les chances de converger vers le minimum global de la fonction soient maximales. Nous avons donc développé une heuristique spécifique à notre problème qui se décompose en deux étapes :

- l'échantillonnage de l'espace des solutions issu des contraintes strictes. Chaque échantillon est évalué au regard de la contrainte souple agrégative.
- la recherche locale à partir d'une ou plusieurs solutions intéressantes sélectionnées suite au résultat de l'échantillonnage.

#### 4.5.2 Echantillonnage de l'espace des solutions

On cherche à évaluer la fonction objectif en couvrant de façon judicieuse l'espace des solutions de façon à ne pas manquer de zone intéressante, mais en limitant le nombre d'évaluations nécessaires au minimum pour que le temps de traitement soit acceptable.

Concernant le placement de la pointe, dans cette phase d'initialisation on considère simplement que la pointe est placée au centre du volume cible. La zone ciblée étant généralement relativement restreinte cette approximation est suffisante pour une évaluation rapide des trajectoires correspondant aux différents points d'insertion possibles.

Les deux paramètres restants concernent l'angle d'insertion de l'outil qui pour un placement fixe de la pointe de l'outil implique un point d'insertion particulier (et réciproquement). Il est alors possible d'échantillonner sur ces deux paramètres en considérant comme point d'insertion potentiel chaque point du maillage représentant la zone d'insertion. Cependant contrairement au volume cible pour lequel les voxels ont tous la même dimension, la zone d'insertion n'est pas forcément maillée de façon régulière. Un nombre élevé d'échantillons proches pourraient alors être sélectionnés sans justification particulière

**Algorithme 4.2** Echantillonnage des angles de placement d'outil

---

```

1:  $L = \emptyset$  {inclinaisons sélectionnées}
2: pour tout point  $P$  de insertionZone faire
3:    $(\alpha, \beta) = coordSpherique(P)$ 
4:    $(\alpha, \beta) = (floorAngleStep(\alpha, step), floorAngleStep(\beta, step)),$ 
5:    $L = L \cup (\alpha, \beta) \cup (\alpha + step, \beta) \cup (\alpha, \beta + step) \cup (\alpha + step, \beta + step)$ 
6: fin pour

```

---

quant à l'intérêt du sous-ensemble concerné, ce qui se traduit par une augmentation du coût en temps de calcul sans répercussion sur la précision du résultat de l'échantillonnage. Pour assurer un échantillonnage régulier sur les angles d'insertion, on discrétise l'espace en choisissant un pas d'angle  $step$ . On limite les angles parcourus à ceux qui correspondent à une insertion dans la zone d'insertion, pour diminuer les évaluations de la fonction objectif au minimum. L'algorithme 4.2 décrit comment on sélectionne les pas d'angle intéressants. La fonction *coordSpherique* calcule les angles correspondant aux coordonnées sphériques du point en paramètre dans le repère  $(O, \vec{x}, \vec{y}, \vec{z})$ ,  $O$  étant le centre du volume cible. La fonction *floorAngleStep* arrondit l'angle au pas d'angle inférieur. L'algorithme consiste donc à ajouter les quatre pas d'angle les plus proches du placement représenté par un point d'insertion dans la liste des échantillons, et ce pour chaque sommet de la zone d'insertion. On aurait également pu envisager un échantillonnage aléatoire dans l'espace de recherche, cependant notre méthode permet de retrouver facilement et rapidement à partir d'un point d'insertion l'échantillon le plus proche en calculant les angles associés au vecteur point d'insertion - point cible. Cette association point d'insertion / échantillon sera utilisée par la suite pour représenter visuellement le respect d'une contrainte dans l'espace de recherche (voir section 5.2.3).

### 4.5.3 Choix des échantillons d'intérêt

Pour chaque échantillon sélectionné la fonction objectif est évaluée et par conséquent c'est le cas également de toutes les contraintes souples dont elle dépend. Les résultats intermédiaires des différentes contraintes souples ainsi que les résultats sur la contrainte aggregative sont stockés pour pouvoir être réutilisés par la suite. Il s'agit alors de déterminer les placements candidats à une exploration plus poussée. On ne peut se contenter de sélectionner simplement l'échantillon correspondant à l'évaluation minimale de la fonction objectif. En effet, il est possible qu'on aboutisse à un meilleur résultat final à partir d'un autre échantillon pour lequel l'évaluation est proche de l'évaluation minimale constatée. L'algorithme 4.3 repère les *vallées* intéressantes dans l'espace des solutions, c'est-à-dire les composantes connexes de l'espace correspondant à une évaluation proche du minimum. Pour cela il sélectionne parmi les triangles de la zone d'insertion, ceux qui correspondent à l'évaluation minimale à  $\epsilon$  près, puis construit les composantes connexes de la zone constitué des triangles d'intérêt (fonction *buildCC*). Ces composantes connexes représentent les vallées de la fonction objectif. Une placement initial est alors sélectionné par composante

**Algorithme 4.3** Choix des placements initiaux de la phase d'optimisation

---

```

1:  $I = \emptyset$  {échantillons sélectionnés}
2:  $minZone = \emptyset$  {triangles d'intérêt de insertionZone}
3:  $min = minEval(L)$  {minimum de la fonction objectif dans l'échantillonnage}
4: pour tout triangle  $T$  de insertionZone faire
5:   si  $eval(T) < min + \epsilon$  alors {évaluation estimée par interpolation sur les échantillons}
6:      $minZone \leftarrow T$ 
7:   fin si
8: fin pour
9:  $CC = buildCC(minZone)$  {construit les composantes connexes}
10: pour tout composante  $C$  de  $CC$  faire
11:    $I \leftarrow placement(center(targetVolume), center(C))$ 
12: fin pour

```

---

connexe afin de lancer une optimisation locale dans la vallée correspondante. Ce placement est constitué du point d'insertion correspondant au meilleur résultat de l'échantillonnage dans la composante et du centre du volume cible pour la localisation de la pointe de l'outil. Cette heuristique part de l'idée qu'il n'y a pas de gros obstacle entre une position initiale quelconque de la composante et la position correspondant à l'évaluation minimale puisque l'évaluation est la même dans toute la composante (à un epsilon près). Un algorithme de recherche locale efficace devrait donc permettre de le rejoindre avec une bonne précision.

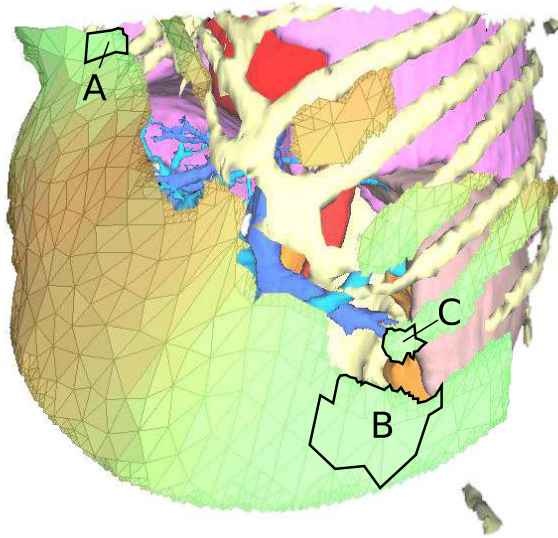


FIGURE 4.11: **Zones d'intérêt pour l'étape d'optimisation locale** : Le résultat de l'échantillonnage de la fonction dans la zone d'insertion est représentée en couleur (dégradé du vert au rouge). Les ensembles  $A$ ,  $B$  et  $C$  sont les composantes connexes sélectionnées pour une optimisation locale.

La figure 4.11 représente le résultat de l'échantillonnage sur un cas particulier et illustre bien la notion de vallée d'intérêt. Le maillage de la zone d'insertion est représenté en dégradé de couleur selon le respect de la contrainte agrégative associé au point d'insertion, allant du vert pour un respect total au rouge pour une infraction maximale. Dans le cas représenté, on repère 2 vallées pour la fonction objectif agrégative considérée. L'une de ces deux vallées est séparée en deux composantes connexes ( $B$  et  $C$ ) dans la zone d'insertion puisqu'une partie de celle-ci ne correspond pas à des stratégies solutions (stratégies traversant les côtes du patient). L'élimination des stratégies non solutions lors de la recherche d'optimum implique un obstacle entre les deux composantes, infranchissable par un processus d'optimisation locale. Les zones  $A$  et  $B$  sont situées dans la même composante connexe de la zone d'insertion, cependant le comportement de la fonction objectif entre les 2 zones ne permettra probablement pas de passer de l'une à l'autre et de les comparer en utilisant une méthode d'optimisation locale. Les valeurs hautes de la fonction objectif entre les deux vallées constituent un obstacle difficilement franchissable. C'est pourquoi seuls les triangles de la zone d'insertion correspondant à un respect quasi-optimal de la contrainte sont sélectionnés et considérés en composantes connexes. Les 3 composantes connexes  $A$ ,  $B$  et  $C$  représentent les vallées d'intérêt. Pour chacune d'elles, un processus d'optimisation local est lancé et aboutit à la au classement de 3 stratégies d'insertion. La comparaison des stratégies issues des 3 optimisations locales permet de déterminer un minimum avec une bien meilleure précision que si l'on se contente d'approfondir uniquement la meilleur stratégie issue de l'initialisation. De plus, cela permet de proposer au praticien les deux stratégies alternatives dont la valeur est quasi équivalente.

## 4.6 Validation de l'heuristique sur un exemple de fonction objectif

Nous avons proposé une heuristique spécifique permettant de répondre à notre problème d'optimisation. Comme dans la plupart des cas d'optimisation dite "difficile", il n'est pas possible de garantir que la solution vers laquelle l'heuristique converge est effectivement l'optimum global de la fonction considérée. On souhaiterait tout de même pouvoir avoir une idée de l'efficacité de l'heuristique. Nous avons donc voulu vérifier les résultats de l'heuristique sur une fonction objectif particulière correspondant à la contrainte souple de minimisation du volume détruit par radiofréquence :

$$obj(X) = \begin{cases} volume(toolEffect(X)) & \text{si } cross(toolTrajectory(X), insertionZone) \\ max & \text{sinon} \end{cases}$$

Cette fonction est intéressante à observer pour plusieurs raisons. Elle est relativement régulière dans le sens où, en dehors des zones de l'espace de recherche où une pénalité est affectée, l'amplitude des valeurs considérées est limitée. Cette caractéristique se retrouve dans la plupart des fonctions considérées pour la planification d'outil : en effet la valeur d'une contrainte considérée pour la planification d'un placement d'outil de chirurgie ne se met pas à changer considérablement pour une modification minime du placement. De plus

dans le cas contraire, on pourrait mettre en doute l'optimalité d'un tel placement qui, sujet à une petite imprécision de reproduction inévitable dans la réalité, pourrait correspondre à une augmentation dramatique de la fonction de coût correspondante. La fonction objectif considérée est aussi intéressante dans le sens où elle présente de nombreux minima locaux, introduits en grande partie grâce à la pénalité appliquée aux placements situés en dehors de la zone autorisée. Cette fonction semble donc être une bonne candidate pour avoir une idée de l'efficacité de notre heuristique sur un exemple concret.

### 4.6.1 Algorithme quasi-exhaustif pour le calcul de l'optimum global de la fonction étudiée

Pour estimer la précision du résultat de notre algorithme d'optimisation, il est nécessaire de pouvoir le comparer avec le résultat attendu et donc le calculer de façon sûre. Il n'existe pas de méthode garantissant la convergence vers le minimum global d'une fonction quelconque dans un domaine borné continu. Par contre si on se place dans un domaine discret borné, le nombre de possibilités à étudier est fini, et donc en les comparant toutes la détermination de l'optimum global est garantie. En discrétisant suffisamment l'espace des paramètres il est alors possible de garantir de trouver le placement correspondant au minimum global théorique de la fonction objectif avec une précision dépendant de la finesse de la discrétisation. Cependant l'évaluation de la fonction objectif sur une discrétisation suffisamment fine et complète de l'espace de recherche (à 5 dimensions) est très coûteuse en temps de calcul et n'aurait pu être effectuée sur plusieurs exemples en un temps raisonnable. Dans un premier temps nous montrerons que la fonction à laquelle on s'intéresse n'est sujette aux minima locaux que sur une partie de ses paramètres. Cette propriété nous permet de proposer une méthode quasi-exhaustive de calcul du minimum théorique de la fonction objectif fournissant un résultat dans un délai acceptable. Nous comparerons alors les résultats de notre heuristique avec ceux fournis par la méthode quasi-exhaustive.

#### 4.6.1.1 Elimination du problème des minima locaux en fixant une partie des paramètres

Nous allons montrer que, pour un axe d'aiguille fixé, la fonction qui associe chaque placement de la pointe au volume de la lésion correspondant au placement et incluant le volume à traiter ne comporte d'autre minimum que le minimum global, quel que soit l'axe fixé, quelle que soit la tumeur traitée.

Dans le cadre de notre travail sur la planification de placement d'outil pour la RFA nous avons assimilé la zone de nécrose induite à un sphéroïde dont le rapport entre grand axe et petit axe (coefficient  $k$ ) est fixe et dépendant du matériel utilisé. Le centre de ce sphéroïde se situe à la pointe de l'aiguille et son axe principal correspond à celui de l'outil. Considérons un repère orthonormé  $(O, x, y, z)$  tel que l'axe  $x$  correspond à la direction fixée de l'axe de l'aiguille. L'équation de la famille de sphéroïdes pouvant représenter la lésion



est la suivante :

$$\frac{(x - x_p)^2}{r^2} + \frac{(y - y_p)^2}{k^2.r^2} + \frac{(z - z_p)^2}{k^2.r^2} = 1$$

où  $P(x_p, y_p, z_p)$  est le centre du sphéroïde et donc la position de la pointe de l'aiguille.

Supposons que le point  $P$  se déplace sur une ligne  $l$  quelconque définie de façon paramétrique par :

$$f(t) = \begin{cases} x(t) = a_0 + b_0.t \\ y(t) = a_1 + b_1.t \\ z(t) = a_2 + b_2.t \end{cases}$$

On considère alors la fonction  $f_Q$  qui, à chaque point  $P$  appartenant la droite  $l$ , associe le rayon du sphéroïde minimal de notre famille de sphéroïdes centrés en  $P$  et englobant le point  $Q$  appartenant au domaine de la tumeur ( $\mathcal{T}$ ), nous obtenons :

$$f_Q^2(t) = (x_Q - (a_0 + b_0.t))^2 + \frac{(y_Q - (a_1 + b_1.t))^2}{k^2} + \frac{(z_Q - (a_2 + b_2.t))^2}{k^2}$$

La courbe  $\Gamma_Q$  qui représente graphiquement la fonction  $g_Q = f_Q^2$  est une parabole dont la concavité est tournée vers le haut. Il existe donc une unique valeur  $t_Q$  pour laquelle  $g_Q(t)$  est un minimum, ce qui signifie qu'il existe une seule position de la pointe de l'aiguille sur  $l$  pour laquelle la nécrose (contenant  $Q$ ) est de taille minimale. Pour un point  $P$  fixé sur  $l$  (de paramètre  $t$ ) et un point  $Q$  de la tumeur, le rayon de tous les sphéroïdes englobant  $Q$  est supérieur à  $g(t)$ . Lorsque  $t$  varie dans  $R$  et donc  $P$  varie sur  $l$ , les rayons de tous les sphéroïdes englobant  $Q$  sont formés de l'enveloppe convexe  $\mathcal{C}(\Gamma_Q)$ . Les sphéroïdes qui englobent tous les points  $Q$  de la tumeur sont représentés par l'intersection  $\cap \mathcal{C}(\Gamma_Q)$  de toutes les parties  $\mathcal{C}(\Gamma_Q)$  du plan se trouvant au dessus des courbes  $\Gamma_Q$ . L'intersection quelconque de convexes étant convexe, on peut conclure que  $(\cap \mathcal{C}(\Gamma_Q))$  est également une partie convexe du plan. Le bord de cet ensemble représente la fonction convexe  $g_{\mathcal{T}}(t) = f_{\mathcal{T}}^2(t) = \text{Max}_{Q \in \mathcal{T}}(f_Q^2(t))$  représentant le rayon du sphéroïde minimal englobant la tumeur  $\mathcal{T}$  associé à un point de placement d'aiguille. Or pour toute fonction convexe non constante, tout minimum local est nécessairement un minimum global.

Cette propriété que nous venons de démontrer est illustrée par la figure 4.12(b).  $P_1$  et  $P_2$  sont les centres des sphéroïdes minimaux englobant respectivement  $Q_1$  et  $Q_2$ . Ils sont définis paramétriquement sur la droite  $l$  par  $t_1$  et  $t_2$ . Dans le cas le plus simple où  $\mathcal{T}$  est réduit à deux points :  $\mathcal{T} = \{Q_1, Q_2\}$ , la fonction  $g_{\mathcal{T}} = \text{Max}(g_{Q_1}, g_{Q_2})$  est formée de deux branches de paraboles se raccordant au point  $S$  d'abscisse  $t_0$ , valeur du paramètre du point  $P_0$ , le centre du sphéroïde minimal englobant à la fois  $Q_1$  et  $Q_2$  (en gras sur la figure 4.12(a)). Si l'on ajoute un troisième point nommé  $Q_3$  à  $\mathcal{T}$ , la parabole  $\Gamma_{Q_3}$  coupe le bord de la région  $\mathcal{C}(\Gamma_{Q_1}) \cap \mathcal{C}(\Gamma_{Q_2})$  en deux points. Le point d'ordonnée minimale, noté  $S'$  fournit le point  $P'_0$ , centre du sphéroïde minimal englobant les trois points  $Q_1$ ,  $Q_2$  et  $Q_3$ . Dans l'exemple représenté sur la figure 4.12(b) le bord du sphéroïde minimal passe par les points  $Q_1$  et  $Q_2$  comme c'est généralement le cas pour 2 points. En revanche, il est exceptionnel que le sphéroïde minimal passe par tous les points au delà de deux. Dans le cas de la figure 4.12(c), les points  $Q_1$  et  $Q_3$  sont au bord du sphéroïde, alors que  $Q_2$  est à l'intérieur.

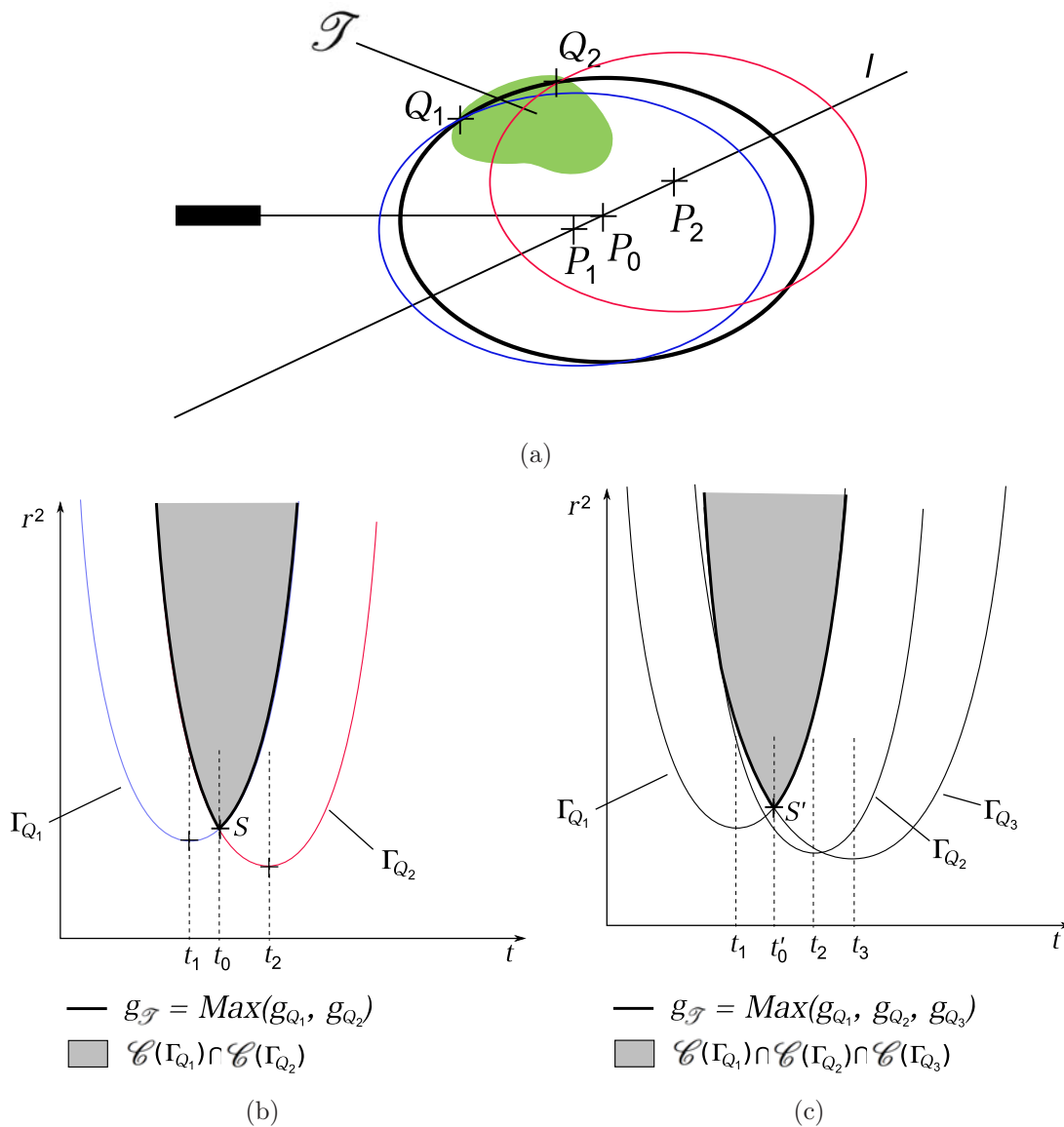


FIGURE 4.12: Convexité de la fonction associant placement d'aiguille et le rayon de la zone de nécrose incluant la tumeur : a) Sphéroïdes de rayon minimal dont le centre se déplace sur la droite  $l$  et englobant les points  $Q_1, Q_2$ , ou les 2 (sphéroïde en gras). b) Représentation graphique des fonctions associant le placement sur la droite au rayon de l'ellisoïde incluant soit  $Q_1$ , soit  $Q_2$ . c) Ajout d'un nouveau point supplémentaire au domaine de la tumeur.

Notre raisonnement nous amène donc à conclure que, pour un axe d'aiguille fixé et si la pointe de l'aiguille se déplace sur une droite quelconque de l'espace, il existe un unique placement d'aiguille qui corresponde à un minimum. Cette propriété peut être étendue à un déplacement quelconque de la pointe dans l'espace. En effet supposons qu'il existe deux placements de la pointe dans l'espace qui correspondent à un volume de nécrose localement minimal, ces deux positions définissent une ligne quelconque de l'espace pour laquelle il existe deux minimums locaux, ce qui contredit ce que nous venons de démontrer. Nous



pouvons donc en conclure que, pour un axe d'aiguille fixé, il existe un unique placement de la pointe dans l'espace pour lequel la nécrose correspondante incluant la tumeur est de taille minimale. Ce placement correspond alors au minimum global de la fonction.

#### 4.6.1.2 Méthode de calcul du minimum théorique

Nous avons montré que dans le cas de la fonction objectif qui nous intéresse, le problème des minima locaux affecte uniquement le choix d'un axe d'aiguille. Cette propriété nous est utile pour le calcul du minimum théorique de notre fonction. Nous avons déjà évoqué le fait que la façon la plus sûre de le trouver est de discrétiser l'espace sur les 5 paramètres concernés et d'effectuer une évaluation exhaustive de toutes les possibilités. Cependant cette méthode est trop lente pour pouvoir fournir un résultat suffisamment précis en un temps raisonnable. Avec une discrétisation de  $3^\circ$  pour les angles et considérant chaque voxel du volume cible comme position potentielle de la pointe, 12 heures de calculs ont été nécessaires pour fournir un résultat sur un exemple concret. Si l'on souhaite multiplier la précision par 2 pour chaque paramètre, le temps de calcul sera lui multiplié par  $2^5$ , ce qui correspond à 384 heures (soit 16 jours) de calcul.

La propriété que nous avons dégagée nous permet de proposer une méthode hybride effectuant une discrétisation (coûteuse en temps de calcul) sur les paramètres concernés par les minima locaux et d'utiliser une méthode d'optimisation locale (convergence rapide) sur les autres paramètres. On discrétise donc les paramètres fixant les angles de l'axe de l'aiguille et pour chaque axe possible (fixé) on utilise une méthode d'optimisation locale (Nelder-Mead) permettant de trouver le meilleur placement d'aiguille correspondant à l'axe fixé. Nous l'avons vu dans la section précédente ce placement de la pointe n'est pas sujet au minima locaux, une méthode d'optimisation locale permettra donc de converger efficacement et rapidement vers l'optimum. En comparant les optima obtenus pour chaque angle possible, on peut fournir de façon sûre une estimation du placement correspondant au minimum théorique. La précision de ce minimum dépend maintenant principalement de la discrétisation sur les deux paramètres représentant l'axe de l'aiguille, l'augmentation de la précision du résultat demande donc une augmentation de temps de calcul beaucoup moins importante. Le temps de calcul de cette méthode hybride se ramène à environ 1h pour une discrétisation avec une précision de  $1^\circ$  sur les angles. Cette méthode ne convient pas pour une optimisation rapide, mais permet de comparer les résultats de notre heuristique d'optimisation avec des résultats assurés, et donc d'évaluer son efficacité et sa précision.

#### 4.6.2 Comparaison des résultats fournis par l'heuristique proposée avec le minimum théorique

Nous avons comparé les résultats obtenus par l'heuristique proposée pour notre solveur de contraintes sur la fonction objectif *obj* définie au début de la section, avec le minimum théorique de cette fonction calculé grâce à la méthode quasi-exhaustive sur 7 cas reconstruits à partir d'images scanner de patient réels. Les résultats de cette comparaison sont représentés dans le tableau 4.1. Tout d'abord remarquons que l'heuristique que nous pro-

posons converge dans tous les cas étudiés en moins d'une minute, alors que le minimum théorique a été fourni en 30 à 60 min. Nous pouvons remarquer que les résultats fournis par les deux méthodes sont vraiment proches. Dans les cas 1 et 4, l'heuristique donne même un résultat légèrement meilleur que la méthode quasi-exhaustive, ce qui peut être expliqué par le fait que cette dernière est limitée par la précision de la discrétisation. Nous pouvons en déduire que notre heuristique fournit des résultats fiables sur la fonction objectif étudiée. Ces résultats laissent penser que celle-ci est également efficace sur des fonctions objectif du même type, c'est-à-dire localement régulières et comportant plusieurs minima locaux, notamment ceux introduits par la pénalité dans les zones interdites de l'espace des paramètres.

TABLE 4.1: Comparaison des résultats de l'heuristique avec le minimum théorique

case #	Heuristique spécifique au solveur		Méthode quasi-exhaustive volume minimal théorique
	temps de calcul (s)	volume minimal (mL)	
1	6	3.07	3.08
2	21	2.49	2.48
3	40	6.84	6.83
4	23	3.83	3.85
5	19	3.17	3.16
6	13	10.62	10.62
7	12	9.32	9.30

## 4.7 Résolution des contraintes de planification : conclusion et perspectives

Les contraintes formalisées dans le chapitre précédent décrivent un problème d'optimisation multi-objectif agrégé sous contraintes. Nous avons présenté les algorithmes développés et les techniques utilisées afin de proposer un solveur efficace sur toutes sortes de contraintes qui peuvent être définies dans le cadre de notre langage. Cela comprend un moteur d'évaluation d'opérations géométriques, un algorithme spécifique pour le traitement des obstacles, une méthode de construction de l'espace des solutions à partir des contraintes strictes, et une méthode d'optimisation de la contrainte souple agrégative. Nous nous sommes également intéressés à l'optimisation d'une fonction objectif particulière, représentative des fonctions objectifs que l'on peut rencontrer. En comparant les résultats de notre heuristique sur cette fonction à ceux obtenus par une méthode garantissant la détermination de l'optimum global, nous avons montré que notre méthode propose des résultats satisfaisants avec une très bonne précision pour un temps de calcul réduit. Les méthodes décrites ont contribué à réaliser un solveur modulaire efficace pour la planification de la stratégie d'insertion d'outil de chirurgie mini-invasive.

Les techniques de résolution décrites dans ce chapitre s'adaptent à n'importe quelles contraintes formalisées dans le cadre défini par le chapitre précédent. C'est-à-dire que la

résolution porte sur le choix d'une stratégie définie par un placement d'outil rectiligne et une zone d'effet associée. Dans le cas d'une opération de radiofréquence, lorsque le volume de la tumeur est trop important il n'est pas possible de la détruire avec une ablation simple. Plusieurs ablations peuvent être alors envisagées. Ce cas deviendra de moins en moins fréquent avec les nouveaux matériels permettant de créer des zones de nécrose de taille de plus en plus importante, cependant il dégage une problématique intéressante : la planification de stratégie concernant plusieurs outils. En réalité, le cas de multiples outils ne pose pas de problème si l'on considère chaque planification comme un problème indépendant. Il suffit de découper la tumeur à traiter en deux volumes (ou plus) et d'appliquer la planification indépendamment sur les différentes masses tumorales pour obtenir une stratégie d'insertion multiple. Cette approche implique un pré-traitement de la part de l'utilisateur qui doit déterminer le découpage de la tumeur pour les ablations multiples (nombre de découpage et plans de coupes). Il serait possible de proposer un découpage automatique en partageant la tumeur en un nombre minimal de volumes assez petits pour pouvoir être détruits par une ablation unique, cependant garantir l'optimalité d'un tel découpage n'est pas une question triviale. De plus cette façon de traiter les différentes stratégies d'insertion séparément les unes des autres interdit d'emblée l'utilisation d'un nouveau type de contrainte : les contraintes traduisant l'interdépendance des différentes stratégies d'insertion. Par exemple, concernant spécifiquement l'ablation par radiofréquence, si une deuxième ablation est nécessaire il faudrait favoriser un placement dans le même axe pour les deux outils pour minimiser le caractère invasif du traitement et si ce n'est pas possible une distance minimum doit être observée entre les deux points d'insertion. La problématique du placement de deux aiguilles de radiofréquence avec contraintes d'inter-dépendance nécessite non plus la résolution de deux problèmes d'optimisations indépendants à 6 paramètres, mais d'un seul problème à 12 paramètres. Une partie de notre travail a été basée sur le fait que l'on connaît le nombre de degrés de liberté du problème et que celui-ci est fixe. Pour intégrer les contraintes d'interdépendance dans la résolution du problème de planification il serait alors nécessaire d'adapter nos méthodes ou de proposer de nouvelles méthodes de résolution adaptées à un nombre variable de paramètres.

Une autre piste de développement de ces travaux concerne le type des données traitées. A l'heure actuelle nous considérons la géométrie des organes comme fixe. En réalité, si cette approximation n'est pas idéale dans le cas des interventions abdominales. En effet, les organes de l'abdomen peuvent se déplacer de plusieurs millimètres à quelques centimètres sous l'effet de la respiration. Pour pallier ce problème, le praticien planifie l'opération sur des images fixes prises à une certaine étape de la respiration et insère l'outil dans le site opératoire à cette même étape de la respiration pour pouvoir reproduire ce geste planifié. Nous avons donc considéré cette même approche pour proposer une planification assistée de stratégie opératoire. Il serait cependant possible là aussi de faire bénéficier l'intervention de techniques de simulation par informatique pour intégrer le facteur de la respiration à la planification. Des travaux effectués à l'Ircad [HNS<sup>+</sup>08] permettent de déduire, à partir d'un nombre d'acquisitions limité, le déplacement des organes de l'abdomen du à la respiration du patient. On pourrait dans un premier temps comparer les résultats obtenus en effectuant

simplement deux résolutions sur chaque cas, une avec des données correspondant à la phase d'expiration, l'autre en inspiration. Il pourrait alors être intéressant d'intégrer ce caractère variable des données dans la résolution de notre problème de planification de façon à évaluer de manière plus précise les stratégies opératoires envisageables.

## Chapitre 5

# RF-Sim : le logiciel de planification spécifique à la radiofréquence

### Sommaire

---

<b>5.1</b>	<b>Structure et implémentation du logiciel de planification . . .</b>	<b>116</b>
5.1.1	Interprétation du fichier de contraintes . . . . .	117
	Interprétation . . . . .	117
	Construction des noeuds internes . . . . .	117
	Construction des noeuds feuilles . . . . .	117
5.1.2	Solveur . . . . .	119
	Evaluation des termes . . . . .	119
	Bibliothèque d'opérations géométriques . . . . .	119
	Application de la planification . . . . .	120
5.1.3	Scène 3D . . . . .	120
5.1.4	Interface . . . . .	121
<b>5.2</b>	<b>Fonctionnalités de l'interface utilisateur . . . . .</b>	<b>121</b>
5.2.1	Visualisation simultanée 2D/3D . . . . .	121
5.2.2	Matérialisation de l'espace des solutions . . . . .	123
5.2.3	Carte des contraintes souples . . . . .	124
<b>5.3</b>	<b>Proposition d'une stratégie optimale : résultats du solveur</b>	<b>127</b>
5.3.1	La stratégie optimale : compromis des différentes contraintes .	127
5.3.2	Performances . . . . .	129
<b>5.4</b>	<b>Adaptation du résultat aux modifications des données . . .</b>	<b>130</b>
<b>5.5</b>	<b>Logiciel de planification : conclusions et perspectives . . . .</b>	<b>133</b>

---

DANS les deux chapitres précédents nous avons présenté deux aspects de ce travail de thèse que sont la formalisation des connaissances issues de l'expertise sous forme de contraintes et la résolution de celles-ci pour fournir une stratégie spécifique de placement d'aiguille. Nous allons maintenant nous intéresser plus particulièrement à l'aspect interaction de notre travail. En effet, dans le processus de conception en CAO les interactions avec le concepteur prennent une place primordiale afin que les échanges entre celui-ci et la machine permettent de converger efficacement vers une solution conforme à ses attentes. Notre démarche tend à s'approcher de ce qui existe en CAO : le praticien est le concepteur auquel on cherche à fournir des outils pour faciliter le travail de planification. Il ne s'agit donc pas de fournir simplement une solution au problème défini, mais également de proposer un certain nombre d'outils permettant de parcourir et visualiser les solutions du problème, et de modifier sa définition si nécessaire. Dans un premier temps nous présenterons la structure interne de RF-Sim et nous aborderons quelques aspects de développement logiciel. Puis nous décrirons les différentes fonctionnalités du logiciel RF-Sim faisant de ce programme un outil complet permettant à l'utilisateur d'accéder à de nombreuses informations sur le cas traité. Nous présenterons également les résultats obtenus par le solveur sur différents patients-test. Enfin nous montrerons comment le logiciel de planification a été développé afin de permettre à l'utilisateur d'intervenir activement dans la planification de la stratégie opératoire.

## 5.1 Structure et implémentation du logiciel de planification

Le logiciel RF-Sim a été développé en C++, en s'appuyant sur des bibliothèques exclusivement multi-plateformes. Le projet complet représente environ 15000 lignes de code réparties dans 44 fichiers sources et 57 classes. La structure du logiciel de planification est représentée sur la figure 5.1. Il est constitué de plusieurs parties :

- l'interpréteur XML qui traduit les règles de planification décrites dans le fichier XML en contraintes utilisables par le solveur.
- la gestion de la scène 3D qui comprend l'anatomie et la pathologie du patient, ainsi que des objets géométriques créés par le solveur pour les besoins de la planification.
- le solveur de contraintes qui est le coeur du logiciel et dont le fonctionnement a été détaillé dans le chapitre 3 du point de vue algorithmique. Il est accompagné d'une bibliothèque d'opérations géométriques permettant d'évaluer les différentes opérations définies par les contraintes sur les objets de la scène.
- l'interface graphique du logiciel qui permet à l'utilisateur de visualiser et manipuler l'anatomie du patient et la stratégie proposée, ainsi que d'accéder à un certain nombre d'informations et d'interagir avec le solveur pour ajuster la stratégie si nécessaire.

Nous présentons sur la figure 5.2 un diagramme de classes simplifié du logiciel. Les classes du diagramme sont positionnées dans l'une ou l'autre des parties que nous venons de décrire selon leur fonction. Certaines classes constituent une interface entre deux parties.

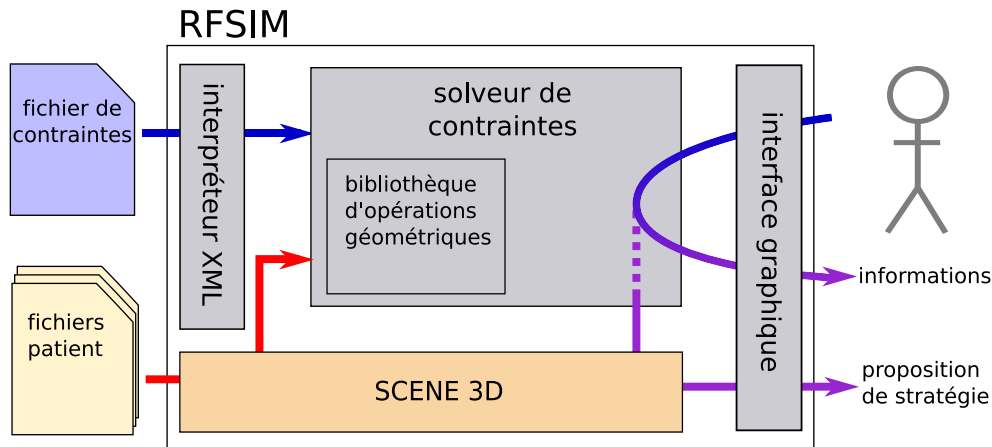


FIGURE 5.1: Structure logiciel de planification d'ablation par radiofréquence

Par exemple la classe *Needle* représente les aiguilles de la scène et constitue un point d'entrée pour la planification, puisque l'objectif est de lui associer une stratégie optimale. La classe *Workspace* n'appartient à aucune zone, son rôle étant essentiellement de regrouper les éléments du contexte et de faire le lien entre le solveur et l'interface du logiciel. Nous nous appuyons sur ce diagramme pour détailler certains aspects de l'implémentation des différentes parties du logiciel.

### 5.1.1 Interprétation du fichier de contraintes

**Interprétation** Au lancement du logiciel l'interpréteur charge le fichier de contraintes par défaut et le traduit sous forme de contraintes. L'analyse lexicale et syntaxique a été codée à l'aide de la bibliothèque Boost Spirit<sup>1</sup>. Elle permet de définir facilement des grammaires et de leur associer des actions sémantiques à l'intérieur de classes C++. Dans RF-Sim, la grammaire du langage de description de contraintes est décrite à l'intérieur de deux classes : *XMLParser* et *OperationParser*. La classe *XMLParser* interprète les parties XML du fichier de contraintes, il crée les contraintes correspondant aux balises XML et leurs champs associés et rattache la contrainte à l'arbre syntaxique créé par la classe *OperationParser*. Cette dernière classe comporte les fonctions *createLeafNode* et *createInternalNode* qui permettent de créer l'arborescence des noeuds associée à la chaîne de caractère lue.

**Construction des noeuds internes** Les noeuds internes sont créés en associant un opérateur avec les noeuds opérands résultant de l'analyse des sous-termes en argument. La fonction statique *makeOperator* de la classe *IOperator* associe les noms des différents opérateurs existants avec la sous-classe correspondante.

**Construction des noeuds feuilles** Les noeuds feuilles stockent la valeur de la donnée associée dans le champ *m\_value*. A l'heure actuelle, les données manipulées sont soit de

1. <http://spirit.sourceforge.net/>

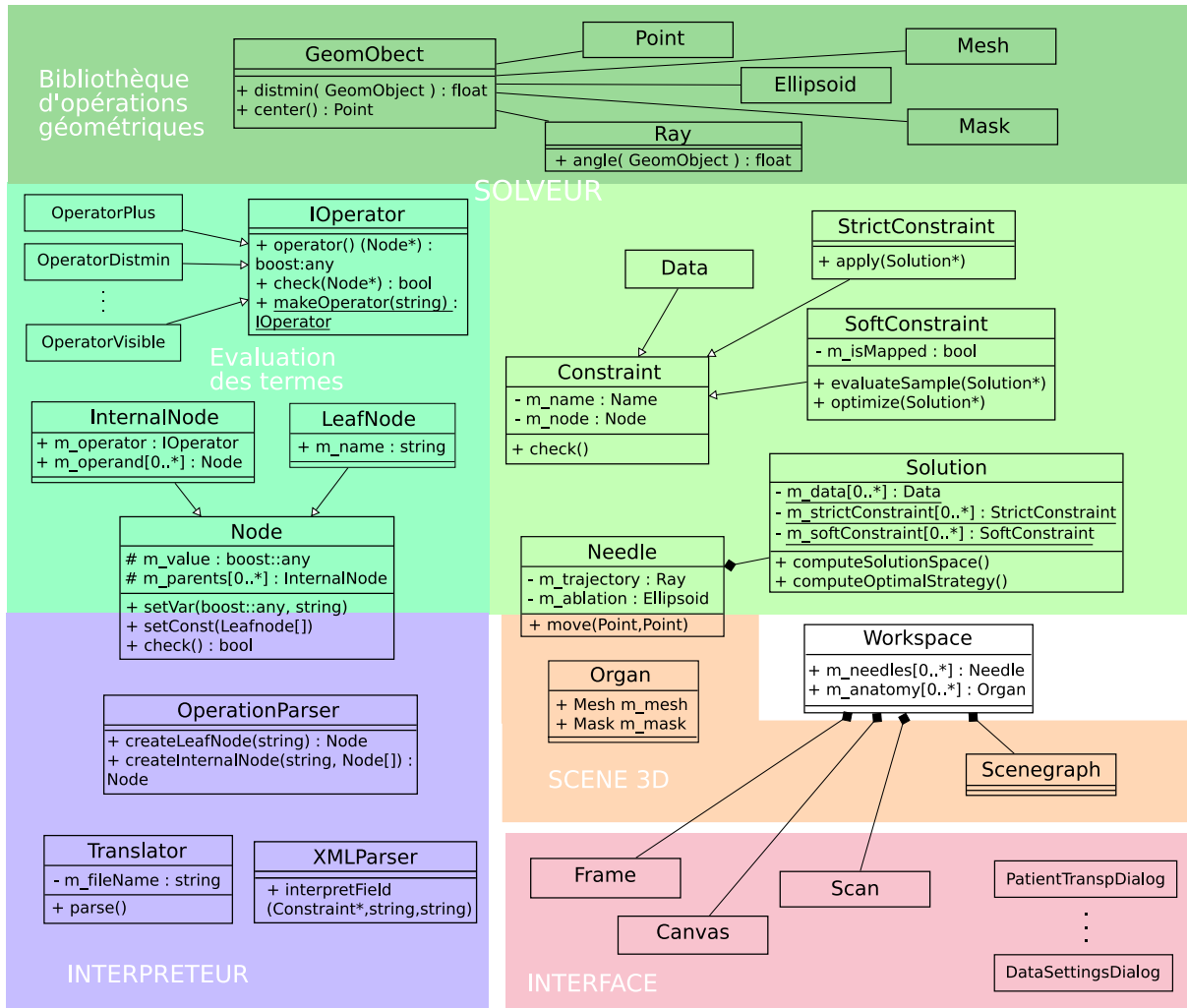


FIGURE 5.2: Diagramme de classes simplifié du logiciel RFSim

type *GeomObject*, *float* ou *bool*. Nous utilisons la classe *any* de la bibliothèque *boost* qui permet de manipuler les données dans les noeuds sans créer de structure particulière et d'envisager de rajouter facilement d'autres types de données dans le futur. La valeur de la donnée associée à une feuille n'est généralement pas accessible au moment de l'analyse du fichier de contraintes. Le nom de la feuille peut désigner un organe ou un noeud inconnu dépendant de la stratégie, celui-ci est alors stocké dans le champ *m\_name* du noeud feuille et la valeur est renseignée dès que l'objet est connu. Lorsque ce nom désigne une donnée supplémentaire ou une contrainte, aucun noeud feuille n'est construit. Le noeud associé est simplement la racine de la donnée ou contrainte correspondante qui a été construite préalablement, lors de l'analyse du fichier. La lecture du fichier donne donc lieu à la construction d'un graphe orienté pour lequel il existe plusieurs entrées correspondant aux racines des termes associées aux contraintes, ces termes pouvant être liées par leur sous-termes communs. Les contraintes sont alors stockées dans les tableaux statiques de la classe *Solution* : *m\_data*, *m\_strictConstraint* et *m\_softConstraint* qui mettent en commun les contraintes



applicables à toutes les aiguilles considérées.

Une fois le fichier analysé, il est possible de vérifier la pertinence des termes analysés et traduits sous forme de contraintes grâce à la fonction *check* de la classe *Constraint*. Celle-ci transmet la requête dans l'arbre associé au terme, de façon à ce que chaque noeud interne vérifie que le prototype déclaré de l'opérateur est compatible avec les opérandes.

## 5.1.2 Solveur

Le solveur de contraintes est lui même découpé en 3 modules : la bibliothèque d'opérations géométriques, l'évaluation des termes, et le planifieur qui utilise les deux modules précédents pour proposer une stratégie.

**Evaluation des termes** Nous avons appliqué le design pattern *Interpreter* pour cette évaluation (voir section 4.1.1.1). *IOperator* constitue la classe abstraite englobant tous les opérateurs du langage. Si l'on souhaite ajouter un nouvel opérateur, il suffit de créer une classe héritant d'*IOperator* et de définir deux fonctions : le constructeur renseignant le prototype de l'opérateur, ainsi que la fonction *operator()* qui permet d'exécuter l'opération sur les opérandes en paramètres. Le fait de nommer cette fonction *operator()* permet d'utiliser directement l'écriture fonctionnelle : si *plus* est une instance de la classe *OperatorPlus*,  $n_1 n_2$  un tableau de deux noeuds représentant des valeurs numériques, l'appel *plus( n1 n2 )* renvoie le résultat de l'opération  $n_1 + n_2$ . Lorsqu'il s'agit d'une opération complexe entre des éléments géométriques, la fonction *operator()* associée à un opérateur utilise les fonctions de la bibliothèque d'opérations géométriques.

**Bibliothèque d'opérations géométriques** Pour illustrer l'utilisation de la bibliothèque, prenons l'exemple de la résolution de l'opérateur *OperatorDistmin*. Celui-ci implique l'appel de la méthode *distmin* du premier objet géométrique, en prenant le deuxième comme paramètre. Cette méthode est polymorphe, comme d'autres méthodes de la bibliothèque. Cela signifie qu'elle n'a pas le même comportement selon la classe réelle de l'objet géométrique sur lequel elle est appelée. S'il s'agit de calculer la distance entre 2 points, c'est la méthode de la classe fille *Point* qui est appelée lors de la résolution. Cela permet de limiter le nombre d'opérateurs définis, en regroupant des opérateurs sémantiquement similaires existant pour des objets de différentes natures. D'autres opérateurs sont spécifiques à une classe particulière, par exemple : le calcul d'angle entre deux points n'a pas de sens. La méthode *angle* est donc une méthode spécifique à la classe *Ray*, elle peut désigner l'angle entre deux demi-droites ou l'angle entre une forme et une droite au point d'intersection. Notons qu'il est possible lors de l'évaluation que l'opération demandée n'ait pas de sens a posteriori. Dans le dernier exemple, le calcul de l'angle entre une demi-droite et un volume suppose qu'il y ait une intersection entre les deux objets considérés. Dans les cas où l'opération ne peut renvoyer une valeur significative, les fonctions de la bibliothèque lancent une exception, qui est traitée par le solveur en amont.

**Application de la planification** Les classes *StrictConstraint* et *SoftConstraint* implémentent les problématiques de résolution spécifiques respectivement aux contraintes strictes et souples que nous avons détaillées dans les sections 4.4 et 4.5. Les fonctions qui se chargent de la résolution prennent en paramètre une instance de la classe *Solution* qu'elles remplissent avec les résultats. Une *Solution* est associée à chaque aiguille de la scène, chacune d'elles ayant sa propre planification selon la tumeur à laquelle elle est associée. L'instance de *Solution* associée à une aiguille regroupe toutes les données stockées pour la planification, cela comprend la ou les stratégies sélectionnées et ordonnées, mais aussi les ensembles délimitant l'espace des solutions (zone d'insertion, volume cible, volume à traiter), les cartes des contraintes souples, et les évaluations issues de l'initialisation du processus d'optimisation. C'est également cette classe qui se charge de transmettre les informations de planification visuelles à la scène 3D (classe *Scenegraph*) à chacune des étapes de résolution.

### 5.1.3 Scène 3D

Les fichiers en entrée sont au format *inr.gz* pour les données volumiques issues de la segmentation, et dans un format utilisé localement pour les données surfaciques. Ces deux types de fichiers sont issus du logiciel de reconstruction développé à l'IRCAD qui permet de travailler sur les images DICOM issues de l'acquisition scanner pour construire les modèles 3D des organes du patient. La visualisation des données patient à travers le logiciel RF-Sim repose donc sur un travail préalable de segmentation et reconstruction des images scanner. La partie visualisation du logiciel repose sur la bibliothèque graphique *vgSDK*<sup>2</sup>. Cette bibliothèque repose sur la définition d'un graphe de scène pour la visualisation d'objets 3D (voir figure 5.3). La lecture d'un tel graphe suit l'ordre du parcours en profondeur. Les objets de la scène sont rassemblés dans différents groupes ou sous-groupes du graphe, qui permettent d'activer ou de désactiver la visualisation de certaines parties de la scène. Les noeuds matériaux définissent une couleur et une transparence pour les objets. Les noeuds de transformation permettent de déplacer ou déformer les objets dans la scène. Les noeuds matériaux ou de transformation ont généralement une portée globale sur tous les noeuds suivants dans la lecture du graphe. Cependant certains types de noeuds groupe permettent de limiter cette portée. C'est par exemple le cas du noeud *needle1* qui isole la transformation *needleTransform*, appliqué uniquement au sous-arbre dont *needle1* est la racine. Sur la figure, la structure générale du graphe de scène utilisé pour RF-Sim est représentée. Les organes sont classés sous le noeuds *organs*, sous la forme d'un groupe associant matériau et forme. Les aiguilles, associées à une transformation géométrique représentant leur placement et une zone d'ablation, sont placées sous le noeud *strategy*. Les informations visuelles issues de la planification sont placées dans le noeud *info*, chaque aiguille ayant un sous-groupe dédié. La modification de la scène est effectuée simplement en ajoutant ou en modifiant des noeuds dans le graphe, à travers la classe *Scenegraph*.

---

2. <http://code.google.com/p/vgSDK/>

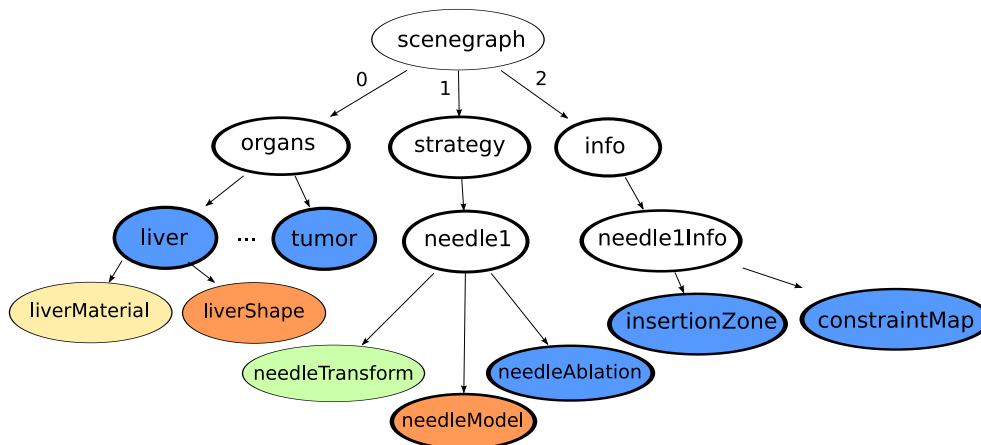


FIGURE 5.3: **Exemple de graphe de scène partiel** : En gris les nœuds groupe, en bleu les nœuds groupe représentant un couple matériau / forme, en jaune un nœud matériaux, en vert les nœuds transformation, en orange les objets géométriques chargés à partir de fichiers de données.

#### 5.1.4 Interface

L'interface du logiciel est développée grâce à la bibliothèque wxWidgets<sup>3</sup> ainsi que certains outils de haut niveau fournis par la bibliothèque vgSDK (manipulation de la scène à la souris, menus contextuels de visualisation prédéfinis, visualisation de coupes scanner). Nous détaillons dans la section suivante les différentes fonctionnalités d'interface proposées par le logiciel du point de vue de l'utilisateur.

## 5.2 Fonctionnalités de l'interface utilisateur

Le rôle du logiciel de planification n'est pas uniquement de proposer une stratégie unique et optimale répondant aux contraintes définies. Il s'agit également de justifier le choix de cette stratégie parmi toutes celles possibles, ainsi que de pouvoir l'adapter voire la modifier complètement, dans les cas où elle ne conviendrait pas pour des raisons qui n'ont pas été prises en compte dans la définition des contraintes. Le logiciel de planification n'a pas pour vocation de prendre la place du médecin et proposer une stratégie à suivre sans remise en cause possible. Il doit plutôt être considéré comme une assistance pour le praticien, lui donnant accès à un grand nombre d'informations afin d'ajuster la planification et valider la stratégie à mettre en oeuvre en salle d'opération.

### 5.2.1 Visualisation simultanée 2D/3D

Les données 3D constituant le patient virtuel peuvent être visualisées dans la fenêtre principale du logiciel. Il est possible de mettre en transparence l'une ou l'autre partie de l'anatomie du patient pour visualiser aisément les régions d'intérêt et de se déplacer autour

3. <http://www.wxwidgets.org/>

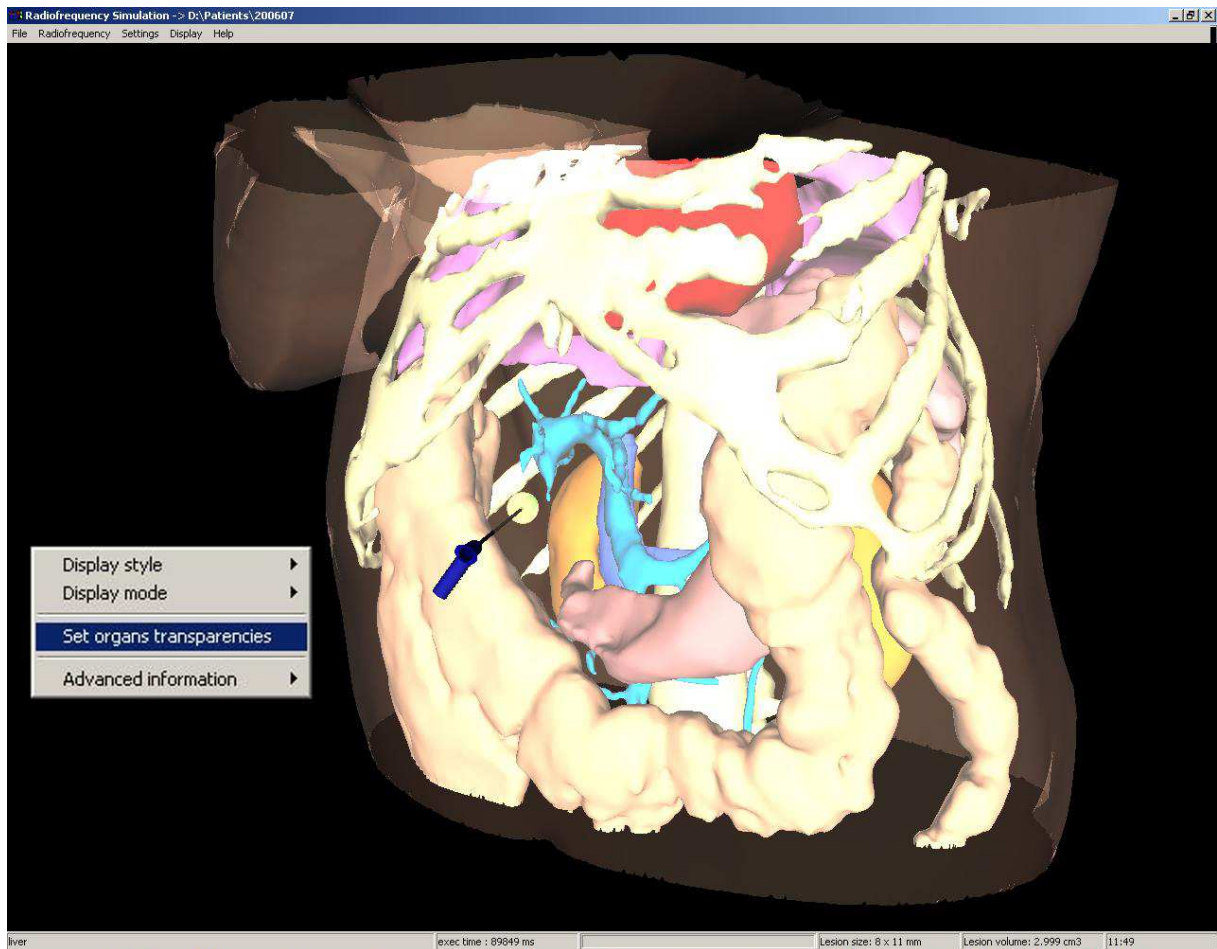


FIGURE 5.4: **Visualisation d'un patient virtuel en 3 dimensions** : Le menu contextuel permet d'accéder à différents modes de visualisation et notamment d'agir sur la transparence des organes de façon à visualiser les organes internes du patient. Ici la peau est en transparence et l'enveloppe externe du foie est totalement omise.

du patient afin d'avoir différents points de vue. Il est également possible de placer une ou plusieurs aiguilles de radiofréquence dans la scène et de la (les) déplacer de façon à visualiser la stratégie d'insertion d'outil que l'on souhaite entreprendre (voir figure 5.4).

Les radiologues interventionnels ont l'habitude de travailler sur des images 2D du patient, que ce soit à travers les images échographiques, CT-scan ou IRM. La visualisation 3D même si elle apporte une meilleure vision globale n'est pas la plus naturelle pour le praticien expérimenté. La visualisation de la trajectoire planifiée à la fois dans une scène 3D et dans les coupes 2D de l'acquisition scanner lui permet alors de se représenter la trajectoire sélectionnée telle qu'il la percevra dans les images per-opératoires qui le guideront pour placer son aiguille conformément à la stratégie planifiée (voir figure 5.5).

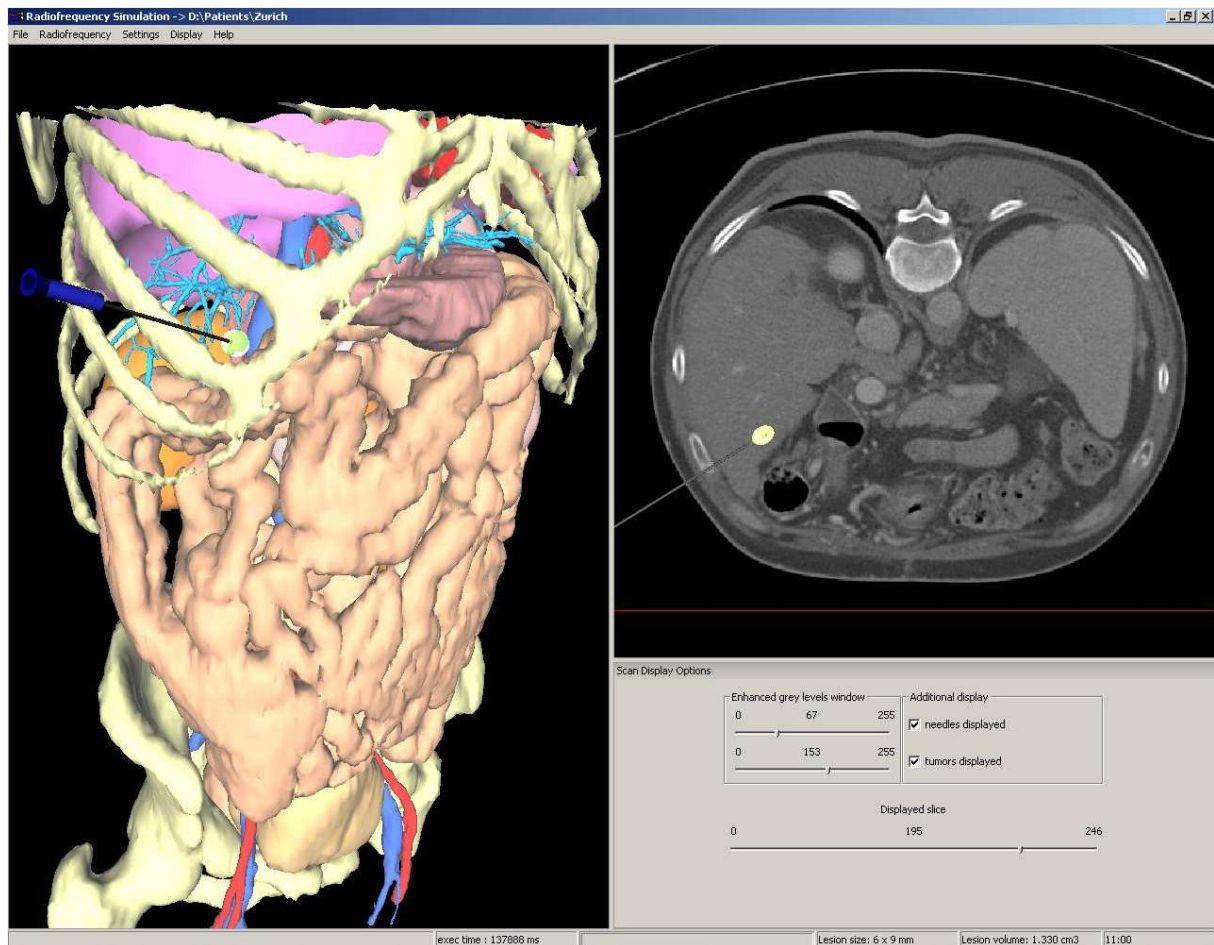


FIGURE 5.5: Visualisation simultanée du patient en 3D et des coupes scanner 2D : La trajectoire planifiée à gauche se retrouve dans les images scanner à droite qui correspondent aux images qui guideront le praticien pendant l'opération.

### 5.2.2 Matérialisation de l'espace des solutions

Dans RF-Sim l'espace des stratégies solutions est représenté par les trois éléments suivants : le volume à traiter, le volume cible et la zone d'insertion. La représentation de cet espace sous cette forme a plusieurs avantages que nous avons déjà évoqués, l'un des principaux étant de pouvoir présenter une information complexe sous une forme visuelle et intuitive pour le praticien. La visualisation des trois éléments définissant l'espace des solutions, et en particulier de la zone d'insertion possible sur laquelle sont appliquées la plupart des contraintes strictes, permet au praticien de visualiser les solutions potentielles d'un seul coup d'oeil. Cette zone est alors matérialisée en vert et en transparence sur la peau (voir figure 5.6).



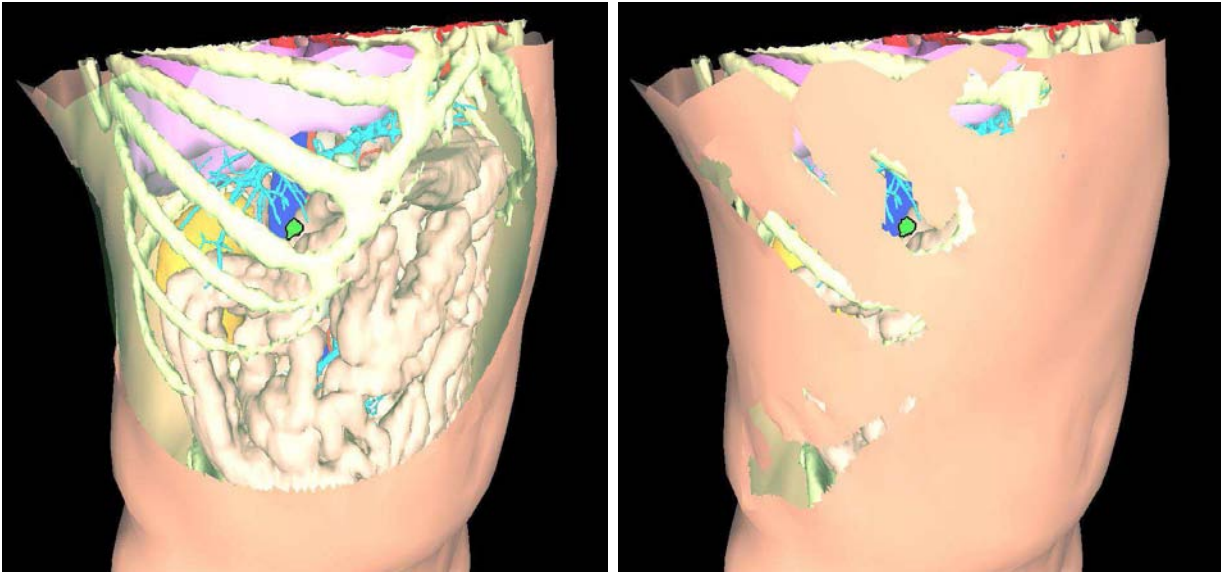


FIGURE 5.6: **Représentation de la zone d'insertion** : La zone d'insertion est en transparence sur la peau, la tumeur qui correspond au volume cible est en vert cerclé de gras. A gauche la zone d'insertion correspondant à la contrainte de longueur d'aiguille, à droite la zone correspondant à la contrainte d'évitement des organes du patient.

### 5.2.3 Carte des contraintes souples

Le logiciel de planification permet d'accéder, à travers un menu contextuel, aux évaluations des différentes contraintes souples correspondant à la stratégie courante. Il est alors possible de parcourir manuellement un certain nombre de stratégies différentes pour pouvoir les comparer vis à vis de la fonction objectif, mais également par rapport à chaque contrainte souple accessible individuellement. Nous avons également proposé une représentation intuitive du respect global des contraintes dans l'espace des solutions, que nous appelons cartes des contraintes souples.

Dans le cas des interventions qui nous intéressent la zone ciblée est, la plupart du temps relativement restreinte et le placement de la pointe est souvent proche du centre du volume cible considéré. L'incertitude quant au choix d'une stratégie d'insertion provient donc en plus grande partie des paramètres concernant le point d'insertion de l'outil. La cartographie des contraintes souples permet alors de visualiser le respect de ces contraintes sur une variation des points d'insertion. En considérant que le placement de la cible lui est fixe, il est possible d'évaluer la valeur d'une contrainte souple pour différents points d'insertion répartis dans la zone d'insertion. On dispose d'un maillage représentant la zone d'insertion à l'issue de la résolution des contraintes strictes. En attribuant une couleur à chaque point du maillage selon que le placement résultant correspond à un respect plus ou moins optimal de la contrainte on peut matérialiser cet échantillonnage de l'espace des solutions. Un dégradé allant du vert pour les solutions les plus respectueuses, au rouge pour les solutions les plus en infraction, permet une visualisation intuitive en accord avec les codes usuels d'utilisation des couleurs. Pour déterminer la couleur associée à une évaluation

**Algorithme 5.1** Détermination de la carte d'une contrainte souple

---

```

1:  $MAP = copy(insertionZone)$ 
2: pour tout point  $P$  de  $MAP$  faire
3:    $(\alpha, \beta) = coordSpherique(P)$ 
4:    $(\alpha, \beta) = (floorAngleStep(\alpha, step), floorAngleStep(\beta, step))$ 
5:    $samples = \{(\alpha, \beta), (\alpha + step, \beta), (\alpha, \beta + step), (\alpha + step, \beta + step)\}$ 
6:    $eval = 0, sumWeights = 0$ 
7:   pour tout  $(a, b)$  de  $samples$  faire
8:      $da = (\alpha - a) / step$ 
9:      $db = (\beta - b) / step$ 
10:     $weight = max(\sqrt{2} - \sqrt{da^2 + db^2}, 0)$ 
11:     $eval += weight * recordedEval(a, b)$ 
12:     $sumWeights += weight$ 
13:  fin pour
14:   $eval /= sumWeights$ 
15:   $setColor(P, eval)$ 
16: fin pour

```

---

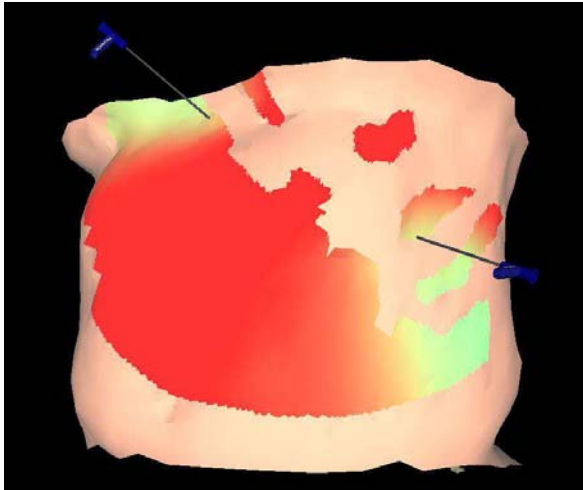
de contrainte, on procède de la façon suivante : pour chaque contrainte souple cartographiée, des valeurs minimale et maximale renseignent l'intervalle de valeur auquel on s'attend pour la contrainte. Une évaluation égale à la valeur minimale correspond à la couleur verte, une évaluation maximale à la couleur rouge et les valeurs intermédiaires permettent d'obtenir une couleur issue du dégradé entre les extrêmes. Selon la façon dont sont définies les contraintes souples considérées, il est possible que le redimensionnement de la contrainte ne garantisse pas des valeurs bornées à l'intervalle  $[minValue, maxValue]$  (cf section 3.4.5.1), ces valeurs extrêmes ne doivent pas perturber la visualisation générale de la contrainte. Une évaluation en dehors de cet intervalle est alors considérée comme équivalente à la borne la plus proche et donc associée à la couleur rouge ou verte selon le cas.

La construction d'une carte de contrainte nécessite de calculer pour chaque point d'insertion une estimation de la valeur de la contrainte correspondante. La multiplication des cartes ne donne pas lieu à une augmentation du temps de calcul nécessaire à la planification complète. En effet, chaque carte est créée à partir de calculs qui sont nécessairement effectués lors de la phase de résolution des contraintes souples. Nous avons vu dans la section 4.5.2 que notre heuristique d'optimisation comporte une phase d'initialisation au cours de laquelle un échantillonnage de la zone d'insertion est effectué, selon un certain pas d'angle. Les évaluations correspondant à cette initialisation sont alors enregistrées et sont utilisées pour estimer la valeur de la contrainte aux points d'insertion correspondant au maillage à colorier. Le dégradé de couleur se fait automatiquement dans les triangles du maillage grâce à l'attribution d'une couleur différente à chaque point des triangles. L'algorithme 5.1 présente le calcul aboutissant à la détermination d'une couleur pour chaque point appartenant au maillage représentant la zone d'insertion. Il est organisé en trois étapes

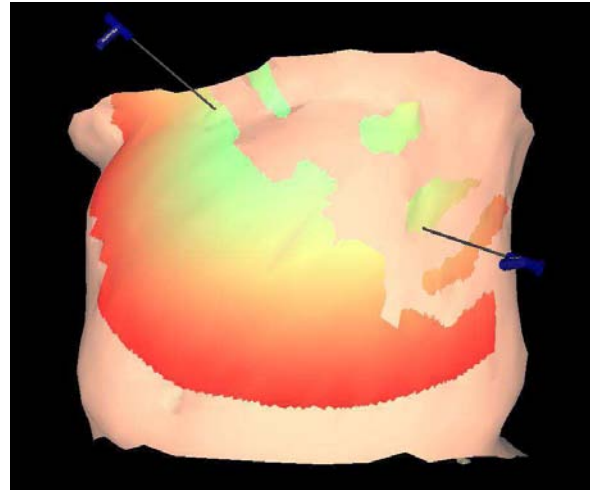
- Déterminer les échantillons les plus proches calculés lors de la phase d'initialisation



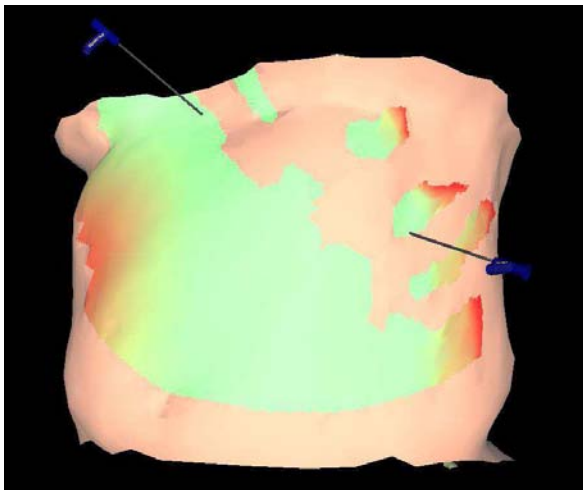
- du processus d'optimisation grâce aux coordonnées sphériques du point considéré (en considérant le centre de la tumeur comme centre du repère).
- Calculer la valeur de la contrainte associée au point d'insertion à partir des valeurs enregistrées pour les échantillons proches. Le poids de chaque échantillon dans l'estimation finale dépend de la distance de l'échantillon au point d'insertion en termes de coordonnées sphériques.
  - Associer une couleur au point du maillage correspondant à la valeur de l'estimation (fonction *setColor*).



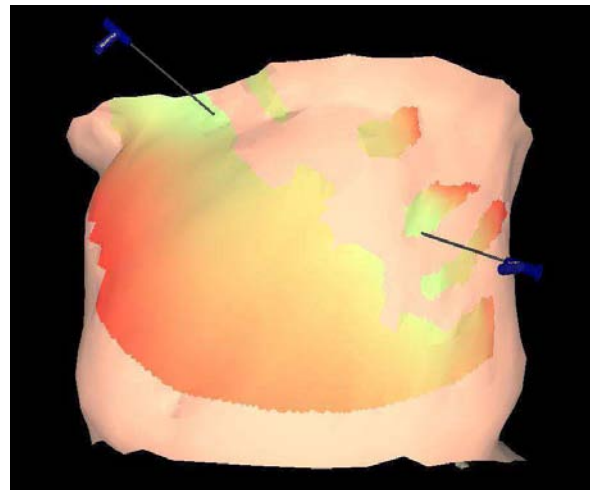
(a) Contrainte de volume minimal de la lésion



(b) Contrainte de profondeur d'insertion minimale



(c) Contrainte de distance aux organes à éviter



(d) Contrainte agrégative optimisée

**FIGURE 5.7: Stratégies proposées par la planification automatique et cartes des contraintes souples associées :** Les deux placements d'aiguille sont les solutions optimales sélectionnées correspondant à un bon compromis des différentes contraintes prises en considération.

Chaque contrainte souple peut donner lieu à la création de la carte correspondante (en option lors de la définition de la contrainte souple, voir section 3.3.3). La contrainte

souple agrégative qui représente la fonction objectif du problème donne automatiquement lieu à la création d'une carte. La figure 5.7 représente les cartes de contraintes issues de la planification pour un patient test reconstruit en 3 dimensions. Concernant la contrainte souple de volume, on distingue 2 pôles sur la carte. Ce cas se présente souvent lorsque l'on a affaire à une tumeur allongée, qu'il vaut mieux atteindre avec un angle proche de celui de son axe principal, de façon à ce que le sphéroïde représentant la zone de nécrose par radiofréquence soit le plus proche possible de la forme du volume à traiter. Une tumeur plus sphérique engendrerait une carte de la contrainte de volume plus uniforme. La carte représentant la contrainte de la profondeur d'insertion se présente sous la forme un dégradé concentrique autour de la position de la peau la plus proche de la tumeur qui correspond au point le plus direct pour atteindre la cible. Dans la carte représentant le risque lié à la proximité des trajectoires aux organes vitaux, les zones rouges sont proches des bords de la zone d'insertion : plus on approche d'un organe plus le danger est grand et lorsqu'on le traverse, la stratégie n'est plus envisageable et n'appartient plus au domaine des solutions. On peut constater que les trajectoires proches de la limite imposée par les côtes sont en vert puisqu'il n'est pas dangereux de frôler les os du patient. Au delà d'une distance de 1 cm pour les autres organes le risque est considéré comme nul, les zones concernées sont donc uniformément vertes. La dernière carte représente la fonction objectif à optimiser qui tient compte à parts égales de chacune des trois contraintes souples citées. Les 2 trajectoires proposées sont donc dans les zones les plus proches du vert de cette carte finale, représentant les meilleurs compromis selon ces trois contraintes.

## 5.3 Proposition d'une stratégie optimale : résultats du solveur

### 5.3.1 La stratégie optimale : compromis des différentes contraintes

Nous avons testé le solveur de contraintes sur différents patients, reconstruits virtuellement en trois dimensions, ayant chacun une ou plusieurs tumeurs à traiter. La planification a été effectuée en tenant compte des contraintes, sélectionnées pour le fichier de contraintes spécifique à la radiofréquence détaillé à la section 3.4. Le résultat principal fourni par le logiciel est la proposition d'une stratégie opératoire optimale répondant à toutes les contraintes strictes et optimisant un compromis des contraintes souples. Lorsque plusieurs stratégies bien distinctes donnent lieu à un compromis de valeurs quasi équivalentes, les différentes stratégies sont proposées dans l'ordre de respect des contraintes souples. Le tableau 5.1 propose un récapitulatif des résultats obtenus pour les 15 tumeurs des 8 patients virtuels dont nous disposons. Ce tableau représente les solutions les plus favorables proposées par le solveur au regard des différentes contraintes souples et les compare avec les valeurs correspondant à un respect optimal de chacune des contraintes. Rappelons que les contraintes souples ont été normalisées de façon à ce que la valeur correspondant à la contrainte soit généralement comprise dans l'intervalle  $[0;1]$ . En observant ce tableau, on remarque que la solution proposée est la plupart du temps à distance suffisante des structures vitales (ne

sont indiqués dans le tableau que les organes à éviter se situant à distance inférieure de la distance de sécurité : 10 mm pour les organes vitaux et 5 mm pour les gros vaisseaux sanguins). En effet, dans la plupart des cas, l'espace des solutions est suffisamment grand pour contenir des stratégies évitant d'approcher ces structures. De même la contrainte de profondeur est le plus souvent très largement satisfaite. Une des contraintes strictes les plus restrictives pour l'espace des solutions est le fait de ne pas pouvoir traverser d'organe vital. Or plus le point d'insertion est éloigné de la cible, plus il y a de chances qu'il y ait un obstacle sur le chemin pour l'atteindre. La zone d'insertion est donc généralement en grande partie constituée de zones proches de la cible qui correspondent à des insertions peu profondes. La contrainte la plus difficile à optimiser est la contrainte de volume, puisque les axes d'insertion correspondant à une zone de nécrose au plus proche du volume à traiter ne sont parfois pas envisageables au regard des contraintes strictes, il est donc souvent nécessaire de faire un compromis plus large sur ce critère. Cependant malgré cela, on remarque que les solutions proposées correspondent à une perte de volume de tissus sains relativement limitée par rapport au volume minimal qu'il est nécessaire de nécroser pour faire une ablation totale du volume à traiter.

cas	contrainte vol.		contrainte risque		contrainte prof.	
	val.	vol. nécrose / <b>min</b> (cm <sup>3</sup> )	val.	prox. organes (mm)	val	prof. insertion/ <b>min</b> (mm)
1	0,21	14,0/ <b>10,8</b>	0,00	aucun	0,02	4,3/ <b>4,1</b>
2	0,01	3,0/ <b>3,0</b>	0,01	veine cave : 5	0,04	3,7/ <b>3,4</b>
3	0,05	2,6/ <b>2,5</b>	0,00	aucun	0,13	5,3/ <b>4,1</b>
4	0,03	2,7/ <b>2,5</b>	0,00	aucun	0,18	5,5/ <b>3,8</b>
5	0,00	9,2/ <b>9,2</b>	0,00	aucun	0,09	5,4/ <b>4,8</b>
6	0,01	10,9/ <b>10,7</b>	0,00	aucun	0,01	4,4/ <b>4,3</b>
7	0,09	5,8/ <b>5,6</b>	0,10	poumon : 9	0,01	4,4/ <b>4,4</b>
8	0,01	6,2/ <b>6,1</b>	0,00	aucun	0,07	4,2/ <b>3,6</b>
9	0,17	3,0/ <b>2,4</b>	0,00	aucun	0,00	5,5/ <b>5,5</b>
10	0,02	3,3/ <b>3,2</b>	0,00	aucun	0,00	5,8/ <b>5,8</b>
11	0,21	4,3/ <b>3,3</b>	0,00	aucun	0,01	5,5/ <b>5,4</b>
12	0,03	3,5/ <b>3,4</b>	0,00	aucun	0,19	7,4/ <b>5,5</b>
13	0,15	5,8/ <b>5,4</b>	0,00	aucun	0,07	6,2/ <b>5,6</b>
14	0,09	9,4/ <b>8,3</b>	0,00	aucun	0,0	8,9/ <b>8,9</b>
15	0,20	4,5/ <b>3,5</b>	0,00	aucun	0,01	4,5/ <b>4,4</b>

TABLE 5.1: Evaluation de la trajectoire résultante au regard des différentes contraintes souples

### 5.3.2 Performances

Comme dans beaucoup de domaines de la médecine, les radiologues interventionnels ont généralement un emploi du temps chargé. Le logiciel RF-Sim a pour objectif d'aider le médecin dans sa planification, cependant il ne faut pas que celle-ci lui fasse perdre un temps précieux. Le logiciel repose sur le fait que les organes représentés sur les images scanner sont déjà étiquetés puis reconstruits en 3 dimensions. Ce processus permettant de récupérer les données indispensables au fonctionnement du logiciel est lui-même coûteux en temps. Il ne repose par contre pas sur le travail du médecin et peut être effectué de façon interactive à l'aide d'un logiciel spécifique par des manipulateurs d'imagerie médicale. En outre il sera de plus en plus automatisé à l'avenir. Pour la planification à travers le logiciel RF-Sim, le médecin doit être présent puisque c'est à lui que revient le choix final d'une stratégie spécifique à l'intervention courante. Il est donc important que ce processus se déroule rapidement, au plus quelques minutes, l'idéal étant qu'il soit interactif. Le logiciel RF-Sim a donc été développé en tenant compte de cet impératif. La redondance des calculs a été limitée au maximum, la bibliothèque des opérateurs a été développée de façon à ce que chaque opération soit la plus efficace possible. La définition du fichier de contraintes a été également effectuée dans cette optique, en formalisant les contraintes à l'aide des opérateurs les moins coûteux ou en limitant la redondance des calculs grâce à la définition de données communes à différentes contraintes.

cas	temps de calcul de la planif. (s)				sans modularité
	calcul esp. sol	init. opt.	opt. locale	total planif.	total planif. (s)
1	107	33	36	<b>176</b>	180
2	28	42	29	<b>99</b>	81
3	25	39	41	<b>105</b>	59
4	34	67	37	<b>138</b>	100
5	62	55	62	<b>179</b>	152
6	51	23	21	<b>95</b>	93
7	30	42	58	<b>130</b>	64
8	26	40	27	<b>93</b>	58
9	13	80	59	<b>152</b>	44
10	14	25	27	<b>66</b>	37
11	12	18	19	<b>49</b>	35
12	16	17	21	<b>54</b>	36
13	19	56	49	<b>124</b>	46
14	25	30	42	<b>97</b>	49
15	33	70	39	<b>142</b>	83

TABLE 5.2: Temps de calcul de la planification automatique proposée par le logiciel RF-Sim

Le tableau 5.2 représente les résultats en terme de temps de calcul de chacune des étapes de la planification. Les tests récapitulés ici ont été effectués sur un Pentium 4 3.2 GHz avec une carte graphique Geforce 7800 GT et 2 Go de mémoire RAM. Le temps total de la planification est comparé au temps total de la même planification avec une version non paramétrable du logiciel. En effet avant de proposer un langage de définition des contraintes de planification, nous avons programmé une première version de RF-Sim dans laquelle la définition des contraintes est codée en dur. La comparaison entre les résultats en termes de temps de calcul de ces deux versions du logiciel nous permet d'évaluer la perte de temps de calcul due à la modularité des contraintes. Les optimisations de calculs possibles dans la première version ne peuvent pas être reproduite dans la version paramétrable, cette dernière étant capable de s'adapter à n'importe quel ensemble de contraintes. Cela a un coût en terme de temps de calcul que l'on peut voir sur le tableau. Le logiciel propose une planification complète en un temps moyen de 111 secondes pour les exemples dont nous disposons contre 73 secondes pour l'ancienne version. Dans les deux cas, on obtient un résultat généralement en 1 à 3 minutes ce qui correspond bien à notre objectif de base. On peut également observer les temps de calculs des différentes étapes de la planification. Le temps nécessaire à la détermination de l'espace des solutions est la partie la plus variable du temps de calcul (de 12 à 107 secondes). Il dépend principalement de la taille du volume cible (la tumeur), puisque la partie la plus coûteuse de la résolution des contraintes strictes est la détermination des zones visibles de la peau depuis la tumeur. Or cette détermination de la visibilité est effectuée depuis une certaine proportion des voxels représentant la tumeur. L'étape d'initialisation du processus d'optimisation dure généralement moins de 1 minute. La longueur de cette étape dépend principalement de l'ampleur de l'espace des solutions. Plus la zone d'insertion est grande, plus il y a d'échantillons à tester pour sélectionner une ou plusieurs stratégies pour l'optimisation locale. Cette dernière étape est la plus stable en termes de temps de calcul. La méthode d'optimisation locale converge en environ une trentaine de seconde vers un optimum. Pour les cas 5, 7 et 9 l'optimisation locale est 2 fois plus longue puisque l'initialisation a abouti à la sélection de deux stratégies à partir desquelles une optimisation locale est effectuée. Les temps présentés dans ce tableau sont évidemment fortement liés aux contraintes qui ont été définies. Plus les opérations décrites par les différentes contraintes nécessitent de puissance de calcul, plus les multiples évaluations nécessaires pour la résolution du problème par le solveur impliquent un temps de planification conséquent. Toutefois remarquons qu'avec un fichier de contraintes décrivant de nombreuses opérations et un ordinateur dont la puissance est modérée, le solveur propose un résultat satisfaisant en un temps raisonnable.

## 5.4 Adaptation du résultat aux modifications des données

Le logiciel de planification permet une visualisation facile de l'anatomie et de la pathologie du patient, ainsi que l'accès à un certain nombre d'informations relatives à l'op-

timisation. L'utilisateur de RF-Sim n'est pas considéré comme passif, il peut également intervenir sur les paramètres de la planification de façon à ce que la stratégie proposée soit plus conforme à ses attentes. Les données définies dans le fichier de contraintes peuvent être accessibles à l'utilisateur à travers une fenêtre permettant de modifier leur valeur par défaut. Il suffit de l'avoir précisé au moment de la rédaction du fichier de contraintes en mettant l'attribut *show* de la donnée à *true*. On aurait pu considérer que toutes les données sont modifiables d'office, cependant certaines d'entre elles ne sont définies qu'à des fins de lisibilité du fichier ou pour limiter la redondance des calculs. Permettre l'accès à ces données ne ferait alors que parasiter l'interface et distraire l'utilisateur quant aux paramètres réglables réclamant son attention. Seules les données booléennes ou numériques sont modifiables de cette manière. En effet non seulement il ne paraît pas aisé de changer les valeurs correspondant à une forme à travers une interface simple, mais nous n'en avons pas trouvé l'intérêt. Il est alors possible de modifier la valeur de ces données avant de demander la résolution des contraintes sur le cas courant, cela ne change pas fondamentalement le fonctionnement du solveur, puisque ce sont seulement les valeurs de noeuds feuilles des différentes contraintes qui sont modifiées et ce sont ces nouvelles valeurs qui seront tout simplement prises en compte lors de la résolution des contraintes.

Si le praticien, après avoir demandé une première résolution du problème sur des données patient particulières, considère que le résultat ne correspond pas tout à fait à ce qu'il attend à cause de l'inadaptation des valeurs des différents paramètres du problème, il lui est alors possible de modifier ces valeurs en fonction de ce qu'il désire. Dans ce cas-là il est inutile de recommencer toute la résolution du problème si les données modifiées n'ont de répercussion que sur une partie des contraintes. Pour diminuer la redondance des calculs après modification d'un élément, un champ a été rajouté à chaque noeud indiquant si la valeur stockée dans le noeud est à jour ou non. Lorsque le solveur résout une contrainte, une valeur est affectée à chaque noeud interne de la contrainte, correspondant au résultat de l'opération dont ce noeud est la racine. Cette valeur est à jour tant qu'aucune modification n'a été effectuée dans le sous-arbre du noeud. Quand l'utilisateur modifie la valeur d'une donnée, les noeuds concernés sont informés d'un changement et savent donc que la dernière valeur calculée n'est pas à jour (mécanisme illustré par la figure 5.8). Lors d'une nouvelle résolution, les calculs à effectuer dépendront alors de l'impact de la modification de la donnée :

- quel que soit le type de la contrainte concernée, un noeud de type constant et dont la valeur est à jour n'est pas réévalué.
- toutes les données dont la racine n'est plus à jour sont réévaluées.
- lorsque le noeud racine d'une contrainte souple n'est pas à jour, toutes les valeurs stockées pour cette contrainte lors de l'initialisation du processus d'optimisation ne sont plus valables. Il faut alors refaire la phase d'initialisation pour toutes les contraintes touchées ainsi que la sélection d'échantillons et la phase d'optimisation locale de la fonction objectif. Les cartes de contraintes concernées sont elles aussi modifiées selon le résultat de la nouvelle phase d'initialisation.
- la résolution des contraintes strictes étant une application successive de filtres sur



l'espace des solutions, à partir du moment où une contrainte stricte n'est plus à jour, le résultat de toutes les contraintes suivantes est affecté, et donc toutes les contraintes à partir de la première contrainte invalidée doivent être réévaluées. De plus, cette nouvelle résolution peut engendrer un espace des solutions plus vaste que l'original, et nécessite donc de réeffectuer une partie du processus d'optimisation. On peut alors considérer que toutes les contraintes souples sont touchées et agir comme décrit au point précédent, à l'exception du fait que, si la contrainte souple est à jour, les valeurs déjà calculées pour les échantillons de l'espace des solutions sont toujours valables.

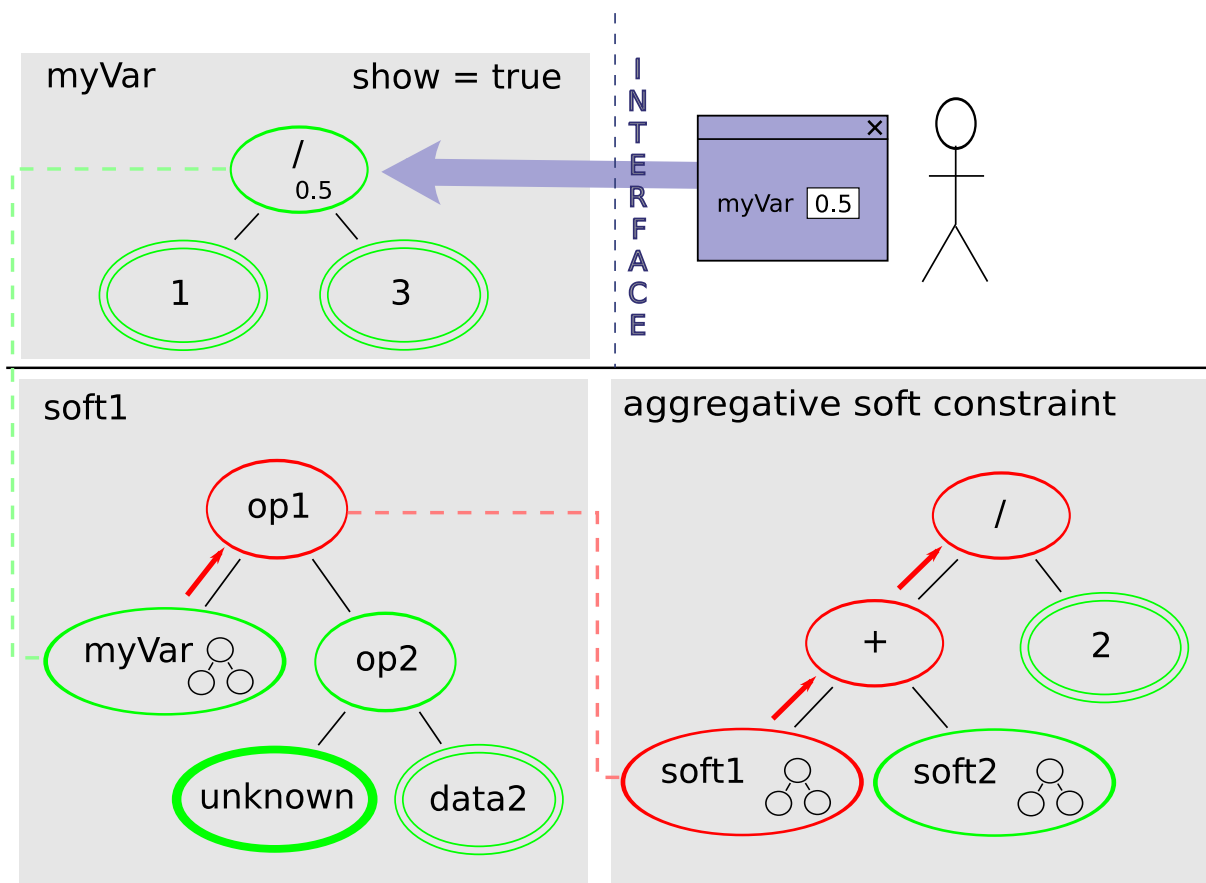


FIGURE 5.8: **Conséquence du changement de valeur d'une donnée sur la résolution des contraintes** : Lorsque la valeur de `myVar` est modifiée par l'utilisateur, la valeur du noeud racine de son terme est changée. La valeur stockée dans le noeud est alors à jour (couleur verte) mais la modification est signalée au noeud racine de la contrainte `soft1` qui la transmet à son tour à ses parents. Tous les noeuds qui contiennent `myVar` dans leur descendance, de façon directe ou indirecte, sont invalidés (couleur rouge)

Prenons comme exemple le fichier de contraintes proposé pour la radiofréquence (voir Annexe). Ce fichier définit plusieurs données accessibles à l'utilisateur : la taille de l'aiguille, l'angle d'insertion minimal dans le foie, la profondeur minimale de la portion de tissus sains du foie à inclure dans la trajectoire, ainsi que les 3 poids des contraintes souples. Les trois



premières données interviennent sur la définition des contraintes strictes. La modification de l'une de ces données après une première solution proposée par le solveur implique donc de recalculer l'espace des solutions, de refaire la phase d'initialisation dans les zones de l'espace des solutions qui n'étaient pas couvertes lors de la première évaluation, et de réeffectuer une phase d'optimisation sur la fonction objectif à partir des stratégies les plus favorables. Si l'on a une modification à faire sur la valeur par défaut de ces données, il est préférable de l'effectuer avant de demander la résolution du problème par le solveur. Les trois données de poids des contraintes souples interviennent uniquement sur la contrainte agrégative qui représente la fonction objectif du problème. Une modification de l'une de ces données après une première planification implique uniquement de recalculer la phase d'initialisation pour la contrainte agrégative à partir des valeurs stockées pour les contraintes souples dont elle dépend. Ce calcul est quasi immédiat. La carte associée représentant la fonction objectif peut donc être visualisée directement après la validation des nouvelles valeurs des données. Il est alors possible de recalculer la trajectoire associée aux nouveaux poids grâce à une simple phase d'optimisation locale. Plus la donnée intervient tard dans le processus de planification, plus simples sont les calculs nécessaires à la mise à jour de la planification tenant compte des nouvelles valeurs des données. La figure 5.9 illustre l'interface de modification des données complémentaires. Une modification dans le fichier de contraintes implique l'adaptation de l'interface associée, ainsi sur l'exemple les données complémentaires comprennent la longueur de l'aiguille, les poids des contraintes ainsi que le volume de l'ellipsoïde minimal englobant la tumeur. Cette dernière valeur est accessible uniquement à titre informatif. Les valeurs des poids ayant été modifiées par rapport à leur valeur par défaut, la stratégie optimale est recalculée et la carte des contraintes mise à jour dès la validation des nouvelles valeurs.

## 5.5 Logiciel de planification : conclusions et perspectives

Dans le cadre de cette thèse, nous avons conçu et développé un certain nombre d'outils pour la planification d'opérations de chirurgie mini-invasive. Nous les avons ensuite intégrés dans un logiciel spécifique à la planification pour l'opération de radiofréquence percutanée. Ce logiciel a été développé de façon à fournir différents outils au praticien lui permettant non seulement de prendre aisément connaissance du cas à traiter ainsi que de la solution proposée, mais également d'avoir accès à un certain nombre d'informations relatives à la planification de façon visuelle ou textuelle. L'espace des solutions est matérialisé par une zone d'insertion sur la peau. Des cartes de couleurs représentant le respect de chaque contrainte souple dans cette zone sont également visualisables. Nous avons attaché de l'importance à la rapidité de chacune des étapes de planification de façon à ce que le processus complet puisse fournir un résultat satisfaisant en quelques minutes malgré les nombreuses contraintes définies par l'expert. Enfin le logiciel permet à l'utilisateur d'avoir des points d'accès aux paramètres de la planification qu'il peut modifier de façon à adapter la définition des contraintes. Les fonctionnalités fournies par le logiciel sont nombreuses et devraient être profitables à la phase de planification des opérations. Cependant nous

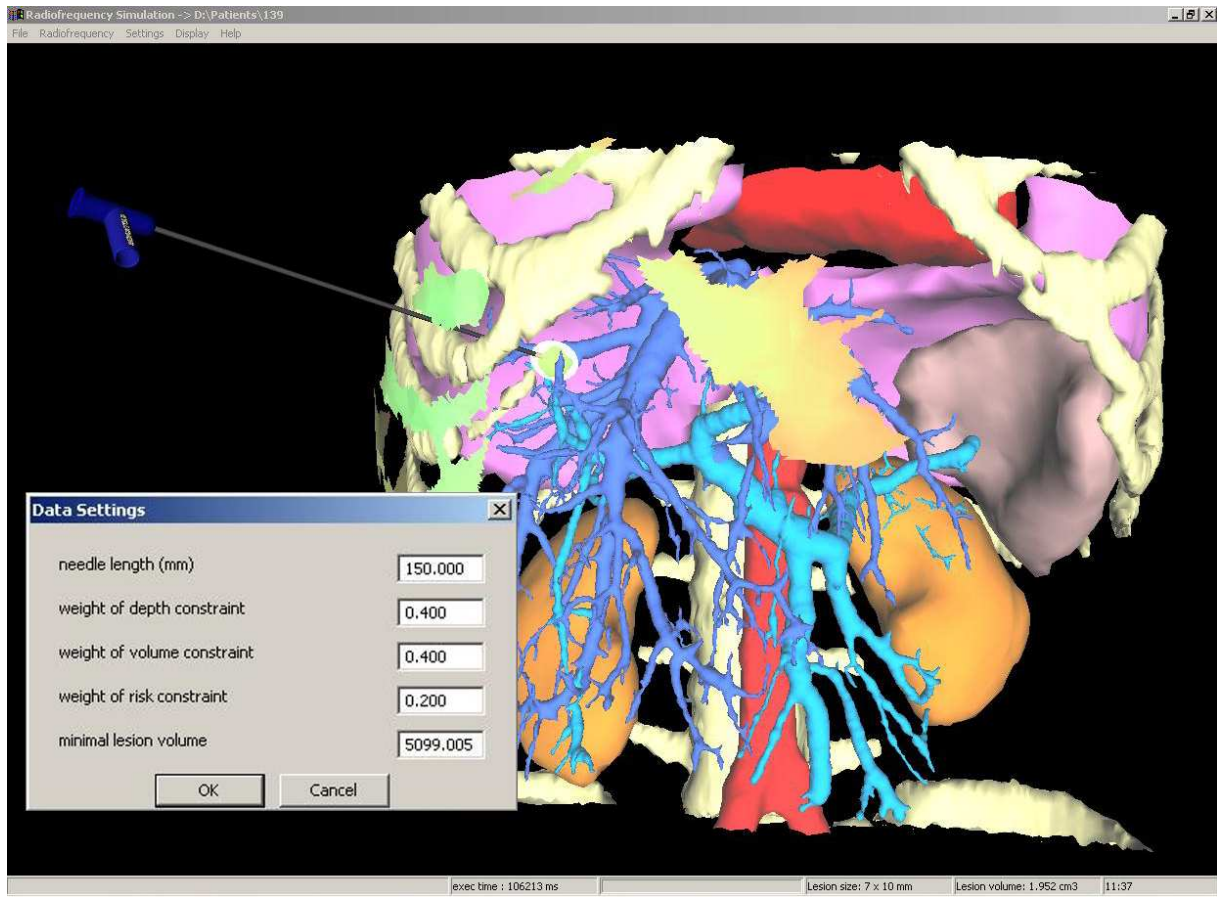


FIGURE 5.9: Modification de la valeurs des données complémentaires

envisageons quelques pistes d'améliorations pour rendre cet outil encore plus complet et pratique pour l'utilisateur.

L'interface du logiciel telle que nous l'avons présentée s'adapte, sur certains aspects, aux contraintes définies dans le fichier de contraintes associé. Pour cela, nous avons introduit dès la formalisation des contraintes des éléments en rapport avec cette interface. Par exemple, certains attributs définis pour les différents types de contraintes se rapportent exclusivement à la visualisation de la contrainte à travers l'interface graphique, c'est le cas de l'attribut *name* mais également de *show* pour les données et *mapped* pour les contraintes souples. On peut imaginer développer cet aspect de personnalisation de l'interface en fonction des contraintes définies.

Certaines données complémentaires sont visualisables et modifiables par l'utilisateur à travers une boîte de dialogue relativement rudimentaire qui indique les noms des données, suivi d'un champ de texte éditable contenant la valeur courante associée. On peut envisager de limiter la marge d'action de l'utilisateur sur ces données ainsi que d'adapter les éléments d'interface les représentant selon leur nature. Certaines données doivent être disponibles uniquement à titre informatif (droit de lecture) alors que d'autres peuvent être modifiées (lecture/écriture). De plus pour celles qui sont modifiables, il pourrait être intéressant

de limiter les modifications à une plage de valeurs. Par exemple les poids des contraintes souples devrait être compris dans l'intervalle  $[0;1]$ . Au niveau de l'interface, la manipulation par curseur semble alors plus adaptée à la représentation d'un intervalle de valeurs qu'un champ éditable.

Le logiciel permet d'accéder aux informations de planification relatives à la stratégie courante. Ces informations regroupent les évaluations de toutes les contraintes souples cartographiées, c'est-à-dire si l'on considère le fichier de contraintes proposé pour la RFA, des valeurs entre 0 et 1. Même si la connaissance de cette valeur donne une bonne indication sur le respect de la contrainte pour la stratégie courante elle ne donne pas d'information concrète. Par exemple au lieu de connaître la valeur de l'indice de la contrainte de volume, on aimerait pouvoir accéder à la valeur du volume de la nécrose. Il faudrait alors pouvoir mettre une sorte d'espion sur un noeud précis du terme indiquant que la valeur de ce noeud est une information intéressante et mérite d'être précisée au même titre que la valeur de la contrainte.

Ces modifications enrichiraient l'interface d'éléments adaptés spécifiquement aux contraintes définies. Elles impliquent cependant qu'un certain nombre d'informations supplémentaires soient précisées dans la définition des contraintes. Il faut garder à l'esprit que cette définition doit rester accessible à un utilisateur peu expérimenté dans le développement logiciel. L'intégration de nouvelles options ne doit pas trop compliquer la formalisation des contraintes par des éléments annexes relatifs à l'interface utilisateur, ou alors il devrait être au moins possible de définir des contraintes cohérentes simples sans obligation de détailler les informations d'interface.

En ce qui concerne les perspectives à plus grande échelle, la planification automatique d'une stratégie ne devrait pas faire l'objet d'un logiciel à elle toute seule. En pratique la multiplication des outils différents pour l'assistance à la planification d'une opération de chirurgie mini-invasive entraîne un alourdissement du processus de planification qui pourrait dissuader le praticien de profiter des différents bénéfices qui peuvent être apportés par l'informatique à différents niveaux de la planification. Afin que le logiciel RF-Sim soit réellement utilisable par le praticien en routine clinique, il faudrait que celui-ci soit intégré dans une plateforme de planification complète intégrant les différentes étapes de planification assistée par ordinateur en un seul outil. Dans l'idéal les différents modules de la plateforme intégreraient :

- segmentation et reconstruction du patient 3D de façon automatique ou assistée.
- planification automatisée d'une stratégie d'intervention sous contraintes, puis sélection d'une stratégie.
- simulation du geste grâce à une interface haptique et simulation approfondie du traitement résultant par intégration de modèles physiques.
- aide à la reproduction du geste dans la salle d'opération grâce à un système de réalité augmentée.



## Conclusion

DANS cette thèse nous avons abordé la problématique de l'assistance à la planification d'opérations de chirurgie mini-invasive de manière originale. Bien que nous nous soyons intéressés particulièrement à l'ablation de tumeurs par radiofréquence, nous n'avons pas orienté nos travaux sur la résolution de la problématique spécifique à cette intervention. Nous avons proposé une approche de résolution générique et adaptable facilement, à la fois à une évolution de la définition du problème de planification, et à la résolution de problématiques similaires. A la manière de certaines approches de conception existant en CAO, le processus de planification est vu sous la forme d'une collaboration entre l'expert qui définit les contraintes propres à l'intervention et le solveur qui est capable de les interpréter et les résoudre sur les données du patient.

Le premier axe de notre travail a donc été de développer des outils de formalisation de la problématique pour l'expert. La définition des contraintes de planification étant relativement fixe d'une intervention à l'autre, nous avons adopté la définition sous forme de script dans un langage simple utilisant le formalisme XML. Les contraintes intervenant dans la planification font intervenir des relations géométriques entre les objets 3D représentant la scène chirurgicale. Le langage associe donc à chaque contrainte la définition d'un *terme* qui représente l'opération à évaluer pour résoudre la contrainte. Ce terme est construit grâce à la combinaison des différents opérateurs, définis afin de pouvoir exprimer un grand nombre de contraintes. Lors de la présentation du langage, nous avons également dégagé deux types de contraintes qui expriment des impératifs d'ordres différents : les contraintes strictes, à respecter, et les contraintes souples, à optimiser. La problématique d'optimisation est définie de manière agrégative, de façon à préciser l'impact des différentes contraintes sur la sélection définitive. Enfin, nous avons utilisé ce langage pour définir un fichier de contraintes, spécifique aux ablations de tumeurs hépatiques par radiofréquence.

Le second axe regroupe les techniques de résolution que nous avons développées pour le solveur de contraintes. L'application de la planification sur un modèle de patient passe par l'interprétation des contraintes définies en XML et la construction des arbres syntaxiques abstraits correspondants. Les évaluations des termes sont effectuées à travers cette structure de données, permettant une manipulation facile et la réutilisation de résultats intermédiaires. Différents opérateurs géométriques ont été développés de façon optimisée, pour permettre une évaluation rapide des termes. En particulier, la contrainte d'évitement de structures critiques, récurrente dans de nombreuses problématiques de planification, nous a incité à proposer un opérateur particulièrement efficace, assimilant le problème de collision à une question de visibilité. Les évaluations des opérations sont utilisées par différents mécanismes du solveur selon la nature des contraintes qu'elles décrivent. La vérification

des contraintes strictes mène à la construction d'un espace des stratégies solutions, par éliminations successives à partir d'un espace initial. La contrainte agrégative décrivant le compromis des différentes contraintes souples est optimisée grâce à une heuristique efficace. Celle-ci comprend une phase d'initialisation, qui repère les régions propices de l'espace des solutions et une phase de recherche locale, qui exploite les stratégies issues de la première phase pour proposer une stratégie optimale avec précision. Cette heuristique a été évaluée pour une fonction objectif représentative, en comparant ses résultats avec des résultats obtenus par une méthode quasi-exhaustive. Elle démontre une très bonne précision pour un temps d'exécution bref.

Le dernier axe sur lequel nous nous sommes penchés implique le développement d'un logiciel de planification d'utilisation intuitive et proposant diverses fonctionnalités d'interaction entre le solveur et l'utilisateur. Les fonctionnalités de visualisation de base comprennent la visualisation du modèle 3D, en parallèle avec les coupes 2D de l'acquisition scanner, le déplacement dans la scène virtuelle, la mise en transparence des structures anatomiques, l'ajout et la manipulation d'aiguilles de radiofréquence. Le logiciel fournit également l'accès à de nombreuses informations de planification : visualisation de l'espace des solutions sous la forme d'une zone en transparence sur la peau, construction de cartes colorées représentant le respect global des contraintes souples, possibilité d'évaluer une stratégie sélectionnée manuellement. La fonctionnalité principale du logiciel est la proposition d'une stratégie optimale associée au fichier de contraintes. Cette fonctionnalité a été testée avec le fichier de contraintes proposé pour la RFA, pour une quinzaine de patients virtuellement reconstruits. Pour les cas considérés, les propositions de stratégies résultantes représentent des compromis satisfaisants pour les différentes contraintes. Enfin le processus de planification assisté implique une certaine adaptabilité et réactivité du logiciel à une modification mineure des contraintes. Le logiciel permet donc l'accès et la modification de certaines données de la planification, définies comme paramétrables lors de la description des contraintes. Cela permet par exemple d'ajuster les poids des différentes contraintes à optimiser et de visualiser de manière interactive les conséquences de l'ajustement sur la planification.

Les travaux présentés dans cette thèse suscitent de nombreuses perspectives à plus ou moins grande échelle, concernant chacun des trois axes de développement que nous avons décrits. De manière plus générale, ces travaux ont ouvert la voie à une nouvelle façon de considérer le processus de planification pré-opératoire à travers une assistance informatique.

D'une part, le fait de considérer les contraintes de planification comme une simple donnée du problème donne au spécialiste la possibilité d'être l'acteur principal de la planification automatisée. Nos travaux n'ont pas pour objectif produire un programme qui effectue la tâche de planification à la place du praticien. Il s'agit de fournir des outils de haut-niveau pour faciliter ce processus, de façon à ce que l'expert puisse se concentrer sur les tâches qui lui sont propres, c'est-à-dire la définition de la problématique et la validation de la stratégie retenue.

D'autre part, alors que la plupart des travaux existant dans le domaine s'intéressent à la problématique de planification d'une intervention particulière, et bien que nous nous

soyons concentrés à l'origine sur la planification de l'ablation par radiofréquence, nous nous sommes attachés à développer des méthodes adaptables et réutilisables pour différents procédés de chirurgie mini-invasive dont les principes sont proches. La généralité de ces travaux pourrait encore être améliorée sur différents aspects :

- Les opérateurs disponibles pour définir les contraintes de planification permettent déjà d'exprimer un grand nombre de choses et l'ajout de nouveaux opérateurs est relativement simple à l'intérieur du solveur. Cependant, chaque ajout nécessite la recompilation du logiciel. Séparer la définition des opérateurs et le logiciel permettrait de faire évoluer plus facilement les possibilités du langage de définition de contraintes.
- A l'heure actuelle, les contraintes décrites dans notre langage peuvent se rapporter à la planification d'une opération basée sur l'insertion d'un outil de type rectiligne associé à une zone d'effet quelconque. En revanche, l'implémentation du logiciel ne permet pas de considérer d'autres outils qu'une aiguille de radiofréquence du type mono-polaire. Là encore l'intégration de nouveaux outils passe par la modification du code spécifique au solveur. Les modèles d'outils ainsi que leur zone d'effet associée, sont pourtant de simples données indépendantes du fonctionnement du solveur en lui-même.
- Enfin il pourrait être intéressant de voir dans quelle mesure les méthodes que l'on a proposées sont généralisables à d'autres traitements pour lesquels la problématique est proche. Nous nous sommes concentrés tout au long du mémoire sur les opérations d'ablation de tumeurs, cependant les possibilités d'applications sont multiples. Citons parmi celles-ci la stimulation cérébrale profonde, pour le traitement de la maladie de Parkinson. On pourrait également envisager d'adapter ces travaux à des procédures plus éloignées du cadre d'origine, impliquant les mêmes types de contraintes, par exemple la radiothérapie, pour laquelle la planification considère le placement et le dosage optimisé de plusieurs faisceaux qui irradient la tumeur tout en préservant au maximum les tissus non ciblés.





# Liste des publications personnelles

## Revue internationale avec comité de lecture

- [1] C. Baegert, C. Essert-Villard, P. Schreck, L. Soler. Trajectory optimization for the planning of percutaneous radiofrequency ablation on hepatic tumors. *Computer Aided Surgery*, 12(2): 82–90, march 2007.

## Conférences internationales avec comité de sélection

- [2] C. Essert-Villard, C. Baegert, P. Schreck. Multi-semantic approach towards a generic formal solver of tool placement for percutaneous surgery. *International Conference on Knowledge Engineering and Ontology Development (KEOD 2009)*, to be published, october 2009.
- [3] C. Baegert, C. Essert-Villard, P. Schreck, L. Soler. Multi-criteria trajectory planning for hepatic radiofrequency. *Medical Image Computing and Computer Assisted Intervention (MICCAI'07), LNCS*, 4792: 676–684, october 2007.
- [4] C. Baegert, C. Essert-Villard, P. Schreck, and L. Soler. Precise determination of regions of interest for hepatic rfa planning. *Studies in health technology and informatics : proceedings of Medecine Meets Virtual Reality (MMVR 15)*, 125:31–36, february 2007.
- [5] C. Essert-Villard, C. Baegert, P. Schreck, L. Soler, A. Gangi. Optimal trajectories computation within regions of interest for hepatic RFA planning. *Medical Image Computing and Computer Assisted Intervention (MICCAI'05), LNCS*, 3750: 49-56, october 2005.



## Bibliographie

- [ACMB00] L. Adhami, E. Coste-Manière, and J.D. Boissonnat. Planning and simulation of robotically assisted minimal invasive surgery. In *MICCAI'2000*, volume 1935 of *Lecture Notes in Computer Science*. Springer, october 2000.
- [AHL<sup>+</sup>02] R. Adam, E.J. Hagopian, M. Linhares, J. Krissat, E. Savier, D. Azoulay, F. Kunstlinger, D. Castaing, and H. Bismuth. A comparison of percutaneous cryosurgery and percutaneous radiofrequency for unresectable hepatic malignancies. *Archive of Surgery*, 137(1332):1339, 2002.
- [AKP<sup>+</sup>06] I. Altrogge, T. Kröger, T. Preusser, C. Büskens, P. Pereira, D. Schmidt, A. Weihusen, and H.O Peitgen. Towards optimization of probe placement for radio-frequency ablation. In *MICCAI'2006*, volume 4190 of *Lecture Notes in Computer Science*, pages 486–493, 2006.
- [AKV<sup>+</sup>02] G. Antoch, H. Kuehl, F. Vogt, J. Debatin, and J. Stattaus. Value of ct volume imaging for optimal placement of radiofrequency ablation probes in liver lesions. *Journal of Vascular and Interventional Radiology*, 13(11):1155–1161, November 2002.
- [Bar03] Vincent Barichard. *Approches hybrides pour les problèmes multi-objectifs*. PhD thesis, Université d'Angers, 2003.
- [BBC<sup>+</sup>09] D. Boltcheva, D. Bechmann, D. Cazier, C. Kern, S. Thery, and P. Schreck. Reconstruction multi-objets d'images 3d multi-labels à partir d'un algorithme de delaunay discret. *Revue Electronique Francophone d'Informatique Graphique (REFIG)*, 3(1):53–65, 2009.
- [BDP99] P. Barral, G. Dorme, and D. Plemenos. Visual understanding of a scene by automatic movement of a camera. In *GraphiCon'99*, 1999.
- [Ber01] Alain Berro. *Optimisation multiobjectif et stratégies d'évolution en environnement dynamique*. PhD thesis, Université des Sciences Sociales Toulouse I, 2001.
- [BGT<sup>+</sup>96] R. Baumann, D. Glauser, D. Tappy, C. Baur, and R. Clavel. Force feedback for virtual reality based minimally invasive surgery simulator. *Studies in health technology and informatics*, 29:564–79, 1996.
- [BP07] D. Bechmann and B. Péroche. *Informatique graphique, modélisation géométrique et animation*. Hermès-Lavoisier, February 2007.

- [BPR04] P.-F. Bonnefoi, D. Plemenos, and W. Ruchaud. Declarative modelling in computer graphics : current results and future issues. In Springer Berlin / Heidelberg, editor, *Computational Science - ICCS 2004*, volume 3039 of *Lecture Notes in Computer Science*, pages 80–89, 2004.
- [BSD<sup>+</sup>00] R. Baissalov, G.A. Sandison, B.J. Donnelly, J.C. Saliken, J.G. McKinnon, K. Muldrew, and J.C. Rewcastle. A semi-empirical treatment planning model for optimization of multiprobe cryosurgery. *Physics in Medicine and Biology*, 45:1085–1098, 2000.
- [BSHW07] C. Basdogan, M. Sedef, M. Harders, and S. Wesarg. Vr-based simulators for training in minimally invasive surgery. *IEE Computer Graphics and Application*, 27(2):54–66, March 2007.
- [BWT<sup>+</sup>00] T. Butz, S.K. Warfield, K. Tuncali, S.G. Silverman, E. Van Sonnenberg, and R. Jolesz, F.A. and Kikinis. Pre- and intra-operative planning and simulation of percutaneous tumor ablation. In *MICCAI'2000*, volume 1935 of *Lecture Notes in Computer Science*, pages 317–326, 2000.
- [CDA99] S. Cotin, H. Delingette, and N. Ayache. Real-time elastic deformations of soft tissues for surgery simulation. *IEEE Transactions On Visualization and Computer Graphics*, 5(1):62–73, January-March 1999.
- [CDMM98] C. Colin, E. Desmontils, J.-Y. Martin, and J.-P. Mounier. Working modes with a declarative modeler. *Comput. Netw. ISDN Syst.*, 30(20-21):1975–1886, 1998.
- [Cha98] L. Champciaux. Classification: a basis for understanding tools in declarative modelling. *Computer Networks ISDN Systems*, 30(20-21):1841–1852, 1998.
- [CMAMB04] E. Coste-Manière, L. Adhami, F. Mourgues, and O. Bantiche. Optimal planning of robotically assisted heart surgery : transfer precision in the operating room. *International Journal of Robotics research*, 23(4):539–548, May 2004.
- [CMG06] C.C. Chen, M. Miga, and R. Galloway. Optimizing needle placement in treatment planning of radiofrequency ablation. In *Proceedings of SPIE*, volume 6141, pages 632–638, 2006.
- [CN04] Isaac Chang and Uyen Nguyen. Thermal modeling of lesion growth with radiofrequency ablation devices. *BioMedical Engineering OnLine*, 3(1):27, 2004.
- [CN05] M. Christie and J.-M. Normand. A semantic space partitioning approach to virtual camera composition. In *Proceedings of the annual eurographics conference*, volume 24-3 of *Computer Graphics Forum*, pages 247–256, 2005.
- [CS01] B. Coyne and R. Sproat. Wordseye : an automatic text-to-scene conversion system. In *Proceedings of SIGGRAPH'01*, pages 487–496, August 2001.
- [CS02] Y. Collette and P. Siarry. *Optimisation multiobjectif*. Eyrolles, 2002.
- [CSD<sup>+</sup>07] S. Clasen, D. Schmidt, K. Dietz, A. Boss, S. Kröber, C. Schraml, J. Fritz, C. Claussen, and P. Pereira. Bipolar radiofrequency ablation using internally

- cooled electrodes in ex vivo bovine liver : prediction of coagulation volume from applied energy. *Investigative Radiology*, 42(1), January 2007.
- [Cur03] A. Curley. Radiofrequency ablation of malignant liver tumors. *Annals of Surgical Oncology*, 10(4):338–347, 2003.
- [DBDW<sup>+</sup>01] T. De Baere, A. Denys, B.J. Wood, N. Lassau, M. Kardache, V. Vilgrain, Y. Menu, and A. Roche. Radiofrequency liver ablation: experimental comparative study of water-cooled versus expandable systems. *American Journal of Roentgenology*, 176(1):187–192, January 2001.
- [Deb07] Laurent Debrauwer. *Design patterns - les 23 modèles de conception : description et solution illustrée en Uml 2 et Java*. Editions ENI, 2007.
- [DFA<sup>+</sup>01] G.D. Dodd, M.S. Frank, M. Aribandi, S. Chopra, and K. Chintapalli. Computer analysis of the size of the thermal injury created by overlapping ablations. *American Journal of Roentgenology*, 177:777–782, October 2001.
- [DPST03] Johann Dréo, Alain Petrowski, Patrick Siarry, and Eric Taillard. *Métaheuristiques pour l'optimisation difficile*. Eyrolles, septembre 2003.
- [EV01] Caroline Essert-Villard. *Sélection dans l'espace des solutions engendrées par un plan de construction géométrique*. PhD thesis, Université Louis Pasteur, Strasbourg, Novembre 2001.
- [Gai03] Véronique Gaildrat. *Modélisation déclarative d'environnements virtuels : Créations de scènes et de formes complexes par l'énoncé de propriétés et l'emploi d'interactions gestuelles. Habilitation à diriger des recherches*. PhD thesis, Université Paul Sabatier, Janvier 2003.
- [Gai07] V. Gaildrat. Declarative modelling of virtual environment, overview of issues and applications. In Dimitri Plemenos and Georges Miaoulis, editors, *International Conference on Computer Graphics and Artificial Intelligence (3IA), Athènes, Grèce, 30/05/2007-31/05/2007*, volume 10, pages 5–15, May 2007.
- [GCB<sup>+</sup>00] A. Gelet, J.Y. Chapelon, R. Bouvier, O. Rouviere, Y. Lasne, D. Lyonnet, and J.M. Dubernard. Transrectal high-intensity focused ultrasound : minimally invasive therapy of localized prostate cancer. *Journal of Endourology*, 14(6):519–528, 2000.
- [GCC<sup>+</sup>03] S. Gaiani, N. Celli, L. Ceciloni, F. Piscaglia, and L. Bolondi. Review article : percutaneous treatment of hepatocellular carcinoma. *Aliment. Pharmacol. Ther.*, 17(2):103–110, 2003.
- [Gol01] S.N. Goldberg. Radiofrequency tumor ablation : principles and techniques. *European Journal of Ultrasound*, 13(2):129–147, June 2001.
- [GP00] V. Gaildrat and D. Plemenos. *Approches déclaratives pour la conception et la modélisation spatiale en synthèse d'images*. 2000.
- [HNS<sup>+</sup>08] A. Hostettler, S. Nicolau, L. Soler, Y. Rémond, and J. Marescaux. *A real-time predictive simulation of abdominal organ positions induced by free breathing*, volume 5104 of *Lecture Notes in Computer Science*. 2008.

- [Hud89] Paul Hudak. Conception, evolution, and application of functional programming languages. *ACM Computing Surveys*, 21(3):359–411, 1989.
- [JS02] P.C. Johnson and G.M. Saidel. Thermal model for fast simulation during magnetic resonance imaging guidance of radio frequency tumor ablation. *Annals of Biomedical Engineering*, 30:1152–1161, 2002.
- [JW00] M.K Jain and P.D. Wolf. A three-dimensional finite element model of radio-frequency ablation with blood flow and its experimental validation. *Annals of Biomedical Engineering*, 28:1075–1084, 2000.
- [KGC98] G. Kwaiter, V. Gaildrat, and R. Caubet. Modelling with constraints : a bibliographical survey. In *Second international conference on Information Visualisation (IV'98)*, page 211, 1998.
- [Kir84] S. Kirkpatrick. Optimization by simulated annealing : quantitative studies. *Journal of Statistical Physics*, 34(5-6):975–986, 1984.
- [KSSH04] Y.S. Khajanchee, D. Streeter, L.L. Swanstrom, and P.D. Hansen. A mathematical model for preoperative planning of radiofrequency ablation of hepatic tumors. *Surgical Endoscopy*, 18:696–701, April 2004.
- [KTN<sup>+</sup>07] Y. Kurumi, T. Tani, S. Naka, H. Shiomi, S. Shimizu, H. Abe, Y. Endo, and S. Morikawa. Mr-guided microwave ablation for malignancies. *International Journal of Clinical Oncology*, 12(2):85–93, 2007.
- [Kud04] M. Kudo. Local ablation therapy for hepatocellular carcinoma: current status and future perspectives. *Journal of Gastroenterology*, 39(3):205–214, 2004.
- [KWOH04] R. Konietschke, H. Weiss, T. Ortmaier, and G. Hirtzinger. A preoperative planning procedure for robotically assisted minimally invasive interventions. In *Jahrestagung der deutschen gesellschaft fur computer und robotassistierte chirurgie (CURAC)*, 2004.
- [LCN99] J.C. Lombardo, M.P. Cani, and F. Neyret. Real-time collision detection for virtual surgery. In *Computer Animation'99*, May 1999.
- [LG05] R. La Greca. *Approche déclarative de la modélisation de surfaces*. PhD thesis, Université de la Méditerranée, Aix-Marseille II, 2005.
- [LGD06] R. La Greca and M. Daniel. A declarative system to design preliminary surfaces. In *WSCG'2006 Full Papers proceedings*, volume 80–86943-03–8, pages 17–24, February 2006.
- [LRGC04] O. Le Roux, V. Gaildrat, and R. Caubet. *Constraint satisfaction techniques for the generation phase in declarative modeling*, pages 194–215. Kluwer Academic Publishers, Norwell, MA, USA, 2004.
- [LSR04] D.C. Lung, T.F. Stahovich, and Y. Rabin. Computerized planning for multi-probe cryosurgery using a force-field analogy. *Computer Methods in Biomechanics and Biomedical Engineering*, 7(2):101–110, 2004.
- [MD01] J.F. McGahan and G.D. Dodd. Radiofrequency ablation of the liver : Current status. *American Journal of Roentgenology*, 176(1):3–16, 2001.



- [MD04] R. Maculet and M. Daniel. Conception, modélisation géométrique et contraintes en cao : une synthèse. *Revue d'Intelligence Artificielle*, 18(5–6):619–645, 2004.
- [MHPA<sup>+</sup>07] L. Maier-Hein, F. Pianka, Seitel A., Müller S.A., A. Tekbas, I. Seitel, Mans Wolf, B.M. Schmied, and Meinzer H.-P. Precision targetin of liver lesions with a needle-based soft tissue navigation system. In *Medical Image Computing and Computer Assisted Intervention (MICCAI2007)*, 2007.
- [MMN<sup>+</sup>02] S. Mulier, P. Mulier, Y. Ni, Y. Miao, B. Dupas, G. Marchal, I. De Weber, and L. Michel. Complications of radiofrequency coagulation of liver tumours. *British Journal of Surgery*, 89:1206–1222, 2002.
- [MNJ<sup>+</sup>05] S. Mulier, Y. Ni, J. Jamart, T. Ruers, G. Marchal, and L. Michel. Local recurrence after hepatic radiofrequency coagulation, multivariate meta-analysis and review of contributing factors. *Annals of surgery*, 242:158–171, 2005.
- [MNM<sup>+</sup>03] S. Mulier, Y. Ni, Y. Miao, A. Rosière, A. Khoury, G. Marchal, and L. Michel. Size and geometry of hepatic radiofrequency lesions. *European Journal of Surgical Oncology*, 29:867–878, 2003.
- [MT09] P. Mathis and S. Thierry. A formalization of geometric constraint systems and their decomposition. *Formal Aspects of Computing*, 21(4), July 2009.
- [Nic04] S. Nicolau. *Un système de réalité augmentée pour guider les opérations du foie en radiologie interventionnelle*. PhD thesis, Université de Nice - Sophia Antipolis, 2004.
- [PJ87] Simon Peyton Jones. *The implementation of functional programming languages*. Prentice Hall, 1987.
- [PNL<sup>+</sup>04] R. Poon, K. Ng, C. Lam, V. Ai, J. Yuen, S. Fan, and J. Wong. Learning curve for radiofrequency ablation of liver tumors : Prospective analysis of initial 100 patients in a tertiary institution. *Annals of Surgery*, 239:441–449, 2004.
- [PTVF02] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C++ : The Art of Scientific Computing (Second Edition)*. Cambridge University Press, 2002.
- [PWP05] T. Preusser, A. Weihusen, and H.-O. Peitgen. On the modelling of perfusion in the simulation of rf-ablation. In *Proceedings of Simulation and Visualisation (SimVis)*, pages 259–268, 2005.
- [RDC<sup>+</sup>04] H. Rhim, G.D. Dodd, K. Chintapalli, B.J. Wood, D.E. Dupuy, J.L. Hvizda, P.E. Sewell, and S.N. Goldberg. Radiofrequency thermal ablation of abdominal tumors : Lessons learned from complications. *Radiographics*, 24:41–52, 2004.
- [RLT04] Y. Rabin, D.C. Lung, and Stahovich T.F. Computerized planning of cryosurgery using cryoprobe and cryoheaters. *Technology in Cancer Research and Treatment*, 3(3):229–243, June 2004.

- [SDHT02] S. Selha, P. Dupont, R. Howe, and D. Torchiana. Dexterity optimization by port placement in robot-assisted minimally invasive surgery. In *Proceedings of SPIE*, volume 4570, pages 97–104, February 2002.
- [SDM01] L. Soler, H. Delingette, and G. Malandin. Fully automatic anatomical, pathological and functional segmentation from ct scans for hepatic surgery. *Computer Aided Surgery*, 6(3):131–142, 2001.
- [SSP<sup>+</sup>99] C.H. Scudamore, I.L. Shung, E.J. Patterson, A.K. Buczkowski, L.V. July, S.W. Chung, A.R. Buckley, S.G. Ho, and D.A. Owen. Radiofrequency ablation followed by resection of malignant liver tumors. *The american journal of surgery*, 177:411–417, 1999.
- [TSH<sup>+</sup>02] S. Tungjitkusolmun, S. Staelin, D Haemmerich, J.Z. Tsai, J.G. Webster, F.T. Lee, D.M. Mahvi, and V.R. Vorperian. Three-dimensional finite element analysis for radio-frequency hepatic tumor ablation. *IEEE Transactions on Biomedical Engineering*, 49(1):3–9, January 2002.
- [WRC<sup>+</sup>00] T.F. Wood, M.D. Rose, D.P. Chung, M. ans Allegra, L. Foshag, and Bilchik A.J. Radiofrequency ablation of 231 unresectable hepatic tumors : indications, limitations and complications. *Annals of Surgical Oncology*, 7(8):593–600, 2000.

# Annexes

## Grammaire du langage XML de définition de contraintes

*CONSTRAINT\_FILE* =

*< constraint\_file >*

*DATA\_DEFINITION*

*STRICT\_CONSTRAINTS\_DEFINITION*

*SOFT\_CONSTRAINTS\_DEFINITION*

*< /constraint\_file >*

*DATA\_DEFINITION* =

*< data\_definition >*

*(DATA)\**

*< /data\_definition >*

*STRICT\_CONSTRAINTS\_DEFINITION* =

*< strict\_constraints\_definition >*

*(STRICT\_CONSTRAINT)\**

*< /strict\_constraints\_definition >*

*SOFT\_CONSTRAINTS\_DEFINITION* =

*< soft\_constraints\_definition >*

*(SOFT\_CONSTRAINT)\**

*< /soft\_constraints\_definition >*

*DATA\_DEFINITION* =  
    < data (XMLFIELD)\* >                    *OPERATION* < /data >

*SOFT\_CONSTRAINT\_DEFINITION* =  
    < soft\_constraint (XMLFIELD)\* >   *OPERATION* < /soft\_constraint >

*STRICT\_CONSTRAINT\_DEFINITION* =  
    < strict\_constraint (XMLFIELD)\* > *OPERATION* < /strict\_constraint >

*XMLFIELD* =  
    *field\_name*    =    " *field\_value* "

*OPERATION* =  
    *operator\_name* ( *OPERATION* (,*OPERATION*)\* )  
    | *value*

## Fichier de définition des contraintes proposé pour la RFA hépatique

```
<constraint_file>
```

```
<data_definition>
```

```
<data name="needle length (mm)" label="needle_length" show="true">
  150
```

```
</data>
```

```
<data name="minimal insertion angle in liver"
  label="tangency_angle" show="true">
  20
```

```
</data>
```

```
<data name="portion of safe liver tissues crossed"
  label="liver_safe_portion" show="true">
  10
```

```
</data>
```

```
<data name="weight of depth constraint"
  label="weight_soft_depth" show="true">
  divide(1/3)
```

```
</data>
```

```
<data name="weight of volume constraint"
  label="weight_soft_vol" show="true">
  divide(1/3)
```

```
</data>
```

```
<data name="weight of risk constraint"
  label="weight_soft_risk" show="true">
  divide(1/3)
```

```
</data>
```

```
<data name="skin" label="skin" obstacle="false">
```

```
</data>
```

```
<data name="liver" label="liver" obstacle="false">
```

```
</data>
```

```
<data name="vital organs" label="organs">
  union( lung, heart, spleen, stomach, kidney, pancreas, intestine )
</data>
```

```
<data name="tumor and its security margin" label="tumor_margin">
  enlarge( tumor, 5 )
</data>
```

```
<data name="volume of tumor margin bounding ball" label="vol_tumorBB">
  volume(boundingball( tumor_margin ))
</data>
```

```
<data name="distance skin/tumor" label="dist_tumor_skin">
  distmin( skin, center( tumor ) )
</data>
```

```
</data_definition>
```

```
<strict_constraints_definition>
```

```
<strict_constraint name= "initial target volume" label="str_1">
  target( tumor )
</strict_constraint>
```

```
<strict_constraint name= "initial treatment volume" label="str_2">
  include( tumor_margin )
</strict_constraint>
```

```
<strict_constraint name= "initial insertion zone" label="str_3">
  cross( skin )
</strict_constraint>
```

```
<strict_constraint name="needle size restriction" label="str_needlesize">
  lower(
    distmin ( center( tumor ), toolInsertionPoint ),
    needle_length
  )
</strict_constraint>
```

```
<strict_constraint name="accessibility restriction" label="str_acces">
  cross(visible( insertionZone, targetVolume ))
</strict_constraint>
```

```

<strict_constraint name="tangency restriction" label="str_tangency">
  lower( tangency_angle, angle( toolTrajectory, liver ) )
</strict_constraint>

<strict_constraint name="liver safe portion garantuee"
  label="str_liverportion">
  lower(
    liver_safe_portion,
    distmin( intersection( toolTrajectory, liver ), tumor )
  )
</strict_constraint>

</strict_constraints_definition>

<soft_constraints_definition>

<soft_constraint name="depth" label="soft_depth"
  mapped="true" minValue="0" maxValue="1">
  divide(
    minus( distmin( toolTip, toolInsertionPoint ), dist_tumor_skin ),
    minus( needle_length, dist_tumor_skin )
  )
</soft_constraint>

<soft_constraint name="volume" label="soft_vol"
  mapped="true" minValue="0" maxValue="1">
  divide(
    minus( volume( toolAblation ), vol_tumorBB ),
    mult(1.4, vol_tumorBB)
  )
</soft_constraint>

<soft_constraint name="risk organs" label="sc_r1" mapped="false">
  max(
    divide( minus( 10 , distmin (toolTrajectory, organs) ), 10 ),
    0
  )
</soft_constraint>

<soft_constraint name="risk vessels" label="sc_r2" mapped="false">
  max(

```



```
        divide( minus( 5 , distmin (toolTrajectory, vessels) ), 5 ),
        0
    )
</soft_constraint>

<soft_constraint name="risk" label="sc_risk"
    mapped="true" minValue="0" maxValue="1">
    plus( sc_r1, sc_r2 )
</soft_constraint>

<soft_constraint name="final" label="soft_final" aggregative="true"
    mapped="true" minValue="0" maxValue="1">
    divide(
        plus(
            mult( weight_soft_depth, soft_depth ),
            mult( weight_soft_vol, soft_vol ),
            mult( weight_soft_risk, soft_risk)
        ),
        plus(weight_soft_depth, weight_soft_vol, weight_soft_risk)
    )
</soft_constraint>

</soft_constraints_definition>

</constraint_file>
```



# PLANIFICATION ASSISTÉE PAR ORDINATEUR POUR LA THERMOABLATION DE TUMEURS

Claire Baegert

L'imagerie médicale et son exploitation par des procédés informatiques permet de fournir de nouveaux outils aux chirurgiens, augmentant leur précision durant les interventions. Cette thèse s'intéresse à la planification préopératoire des interventions de thermo-ablation de tumeurs. L'objectif est de développer des outils intuitifs et automatisés assistant la sélection d'une stratégie opératoire optimale propre à chaque intervention. Nous proposons une approche originale s'inspirant de la modélisation déclarative en CAO. Cette approche considère les règles de planification de base définies par les experts comme des données, qui sont éventuellement modifiables et adaptables selon leur utilisation. Dans cet objectif, nous proposons un langage de définition de contraintes permettant d'exprimer ces règles de façon simple. Nous avons également développé un solveur générique de contraintes permettant de s'adapter aux contraintes définies. Le solveur détecte l'ensemble des solutions adaptées à chaque cas et propose les stratégies d'intervention optimales. Un certain nombre d'outils d'interface sont également développés de façon à fournir au praticien un logiciel de planification intuitif, fournissant des informations précieuses pour le choix de la stratégie opératoire à appliquer.

**Mots clés:** *thermoablation, chirurgie assistée par ordinateur, planification pré-opératoire, contraintes, modélisation déclarative, optimisation multi-objectifs*

## COMPUTER-ASSISTED PLANNING FOR TUMOR THERMAL ABLATION

Medical imaging and its exploitation by computer methods provide new tools for surgeons in order to increase their precision during interventions. This thesis studies the preoperative planning step of tumor thermal ablation. The aim is to develop intuitive and automated tools to assist the selection of an optimal operative strategy, specific to each intervention. We propose an original approach inspired by declarative modelling in CAD. This approach considers basic planning rules defined by experts as data, possibly modifiable and adaptative regarding their use. In this goal, we propose a constraints definition language in order to express them in a simple way. We also developed a generic solver that adapts itself to defined constraints. The solver detects the solutions set associated to each case and proposes optimal intervention strategies. We have also developed some interface tools in order to provide to the surgeon an intuitive planning software, displaying precious information for the choice of an operative strategy.

**Keywords:** *thermal ablation, computer-aided surgery, preoperative planning, constraints, declarative modelling, multi-objective optimization*

