

**République Algérienne Démocratique et Populaire**  
**Ministère de l'enseignement supérieur et de la recherche scientifique**

**Université de Constantine**  
**Faculté des sciences de l'ingénieur**  
**Département : Informatique.**

**Mémoire pour l'obtention du diplôme de magister en informatique**  
**Option : Génie logiciel et Intelligence artificielle**

**Présenté par :**  
**Kadi Soheib**

**Sujet du mémoire :**  
**Exploration Multi-Robots dans des**  
**Environnements Partiellement Connus**

**Soutenu Le : 24 /1 /2009 devant le jury composé de :**

<b>Dr. Salim CHIKHI</b>	<b>Université de Constantine (Président).</b>
<b>Pr. Mohamed BENMOHAMED</b>	<b>Université de Constantine (Rapporteur).</b>
<b>Dr. Azzedine BELAMI</b>	<b>Université de Batna (Examineur).</b>
<b>Dr. Allaoua Chaoui</b>	<b>Université de Constantine (Examineur).</b>

## Résumé:

L'une des tâches importantes dans la recherche en robotique est de permettre à une équipe de robots mobiles d'explorer et de tracer coopérativement un environnement partiellement connu. L'exploration et la cartographie robotique ont de nombreuses applications ou des tâches peuvent être trop dangereuses pour les humains, comme l'espionnage et la détection d'agent chimique ou biologique. L'utilisation de multiples robots pour ces tâches offre une amélioration de l'efficacité et de l'exactitude de traçage.

Ce mémoire se concentre sur le traçage en utilisant les robots mobiles à « capteurs-limités » c.-à-d. des sondes peu coûteuses et à courte portée telles que les sondes infrarouges.

Les robots à capteurs-limités sont relativement peu coûteux à construire, ainsi construire des équipes avec de tels robots est rentable. En plus, en raison de leur coût, les robots à capteurs-limités sont potentiellement jetable dans des applications où les robots pourraient être contaminés par des produits dangereux.

Les deux problèmes de la cartographie et l'exploration multi-robots sont examinés dans ce mémoire. D'abord, un algorithme est développé pour créer les cartes topologiques des environnements d'intérieur rectilignes avec un robot à capteurs-limités. L'algorithme de cartographie présente une méthode pour prolonger la cartographie topologique aux espaces ouverts, longtemps considérée comme un inconvénient de l'approche topologique. Également une approche est présentée pour le problème de « fermeture de boucle » - identifiant quand le robot est revenu à un endroit qu'il a déjà visité.

En second lieu, un algorithme de fusion de deux cartes topologiques sans référence commune est présenté. Cet algorithme emploie les graphes d'appariement et le recalage d'image pour joindre les cartes produites par les différents robots dans une carte globale logique.

Les expériences, exécutées dans la simulation, vérifient l'efficacité des algorithmes du traçage et de la fusion.

# Table des matières

Chapitre 1: Introduction	1
• 1.1 carte topologique	2
Chapitre 2: Les travaux en cartographie et en fusion des cartes	4
• 2.1 la cartographie d'un robot simple	4
• 2.1.1 les cartes de grille d'occupation	4
• 2.1.2 les cartes topologiques	5
• 2.1.3 fermeture de boucle	6
• 2.1.4 robots limités	7
• 2.1.5 les autres travaux	7
• 2.2 la cartographie multi-robot et la fusion des cartes	8
• 2.2.1 les approches	8
• 2.2.2 le graphe d'appariement	9
• 2.2.3 le recalage	10
Chapitre 3 : cartographie avec des robots limités	11
• 3.1 Prétentions	13
• 3.2 l'algorithme de cartographie de base	13
• 3.2.1 comportements	14
• 3.2.2 le processus de cartographie	15
• 3.3 fermeture de boucle	17
• 3.3.1 la construction d'une hypothèse	17
• 3.3.2 vérification d'une hypothèse	18
• 3.3.3 discussion	22
• 3.4 Portails et les couloirs étroits	24
• 3.5 la navigation en utilisant la carte de base	25
• 3.6 améliorations	25
• 3.6.1 les incursions	26
• 3.6.2 Améliorations passive	28
• 3.6.3 Améliorations actives /exploration	28
• 3.7 limites	29
Chapitre 4 : Fusion des topologiques	30
• 4.1 prétentions	32
• 4.2 construction de l'hypothèse	32
• 4.2.1 correspondances des sommets	32

• 4.2.2 augmentation des appariements	33
• 4.2.3 un exemple	34
• 4.3 l'évaluation des transformations et le groupement	36
• 4.3.1 estimation de transformation	36
• 4.3.2 groupement	37
• 4.3.3 suite de l'exemple	38
• 4.4 quelque mots sur l'implémentation	39
• 4.4.1 calculs	40
Chapitre 5 : simulations et résultats	41
• 5.1 la cartographie topologique d'un robot simple	41
• 5.1.1 résultats de la simulation	42
• 5.1.1.1 expériences	43
• 5.2 fusion des cartes topologiques	50
• 5.2.1 résultats de simulation	51
• 5.2.3 discussion	53
Chapitre 6 : conclusions et perspectives	55
• 6.1 la cartographie topologique d'un robot simple	55
• 6.2 fusion des cartes topologiques	56
• 6.3 perspectives	56
ANNEXE A	58
ANNEXE B	62
ANNEXE C	68
BIBLIOGRAPHIE	71

# Liste des figures.

<b>FIG 3.1 : Suivre-mur et détection des caractéristiques.</b>	14
<b>FIG 3.2 : Fermeture de boucle.</b>	16
<b>FIG 3.3 : Les portails et couloirs étroits.</b>	24
<b>FIG 3.4 : Les incursions.</b>	27
<b>FIG 4.1 : Exemple de génération d'hypothèse pour deux cartes d'un monde rectiligne.</b>	35
<b>FIG 4.2 : Les espaces de transformation et les groupes exemple.</b>	39
<b>FIG 5.2 : La cartographie des mondes simples simulés.</b>	44
<b>FIG 5.3 : Construction d'une carte pour un monde « spirale » auto-similaire.</b>	45
<b>FIG 5.4 : Un environnement qui défait la méthode du résonnement par évidence pour la fermeture de boucle.</b>	47
<b>FIG 5.6 : Une carte du monde avec des améliorations incluses.</b>	48
<b>FIG 5.9 : Résultats de fusion des cartes simulées aléatoirement produites.</b>	51
<b>FIG 5.10 : Un appariement entre de grandes cartes.</b>	53
<b>Fig C.1 L'implémentation de EXPLORER.</b>	90
<b>Fig C.2 L'implémentation de DEPLACER-VERS-MUR.</b>	90
<b>Fig C.3 L'implémentation SUIVRE-MUR.</b>	92
<b>Fig C.4 L'implémentation de SUIVRE-COULOIR.</b>	93
<b>Fig C.5 L'implémentation du composant HFO.</b>	94
<b>Fig C.6 L'implémentation de PROBE.</b>	95
<b>Fig C.7 L'implémentation du composant ENTER.</b>	96
<b>Fig C.8 L'implémentation du composant OPP.</b>	96
<b>Fig C.9 L'implémentation du composant EXO.</b>	97

# Liste des tables.

<b>Table C.1: Paramètres des comportements.</b>	77
<b>Table C.2: Les types de sommet dans la carte.</b>	77
<b>Table C.3 : Actions primitives qui sont employées pour construire les comportements de cartographie.</b>	78
<b>Table C.4 Spécification de EXPLORE.</b>	81
<b>Table C.5 DEPLACER-VERS-MUR.</b>	82
<b>Table C.6 SUIVRE-MUR.</b>	83
<b>Table C.7 SUIVRE-COULOIR.</b>	84
<b>Table C.8 HFO(X).</b>	85
<b>Table C.9 PROBE(X,D).</b>	86
<b>Table C.10 ENTER(X).</b>	87
<b>Table C.11 OPP(X,D).</b>	88
<b>Table C.12 EXO(X,D,N).</b>	89

## **CHAPITRE 1**

# **INTRODUCTION**

La construction des cartes en robotique est un problème bien étudié, et les années récentes ont connu une nouvelle croissance pour de divers aspects de la cartographie. Ce mémoire examine deux de ces aspects - le problème de construction d'une carte topologique d'un simple-robot et le problème de fusion des cartes topologique.

Le but à long terme de ce travail est de permettre à des équipes de petits robots

« a capteurs-limités » d'explorer, de construire une carte et de planifier, ces trois tâches doivent être faites d'une façon autonome dans des environnements inconnus. Les capteurs-limités ont comme conséquence sur les robots, une vue limitée de l'environnement.

A la différence des humains, qui peuvent faire la navigation aveugle (sans vision) en employant l'information spatiale précédemment acquise, les robots doivent employer leur capteurs limités pour l'exploration et la cartographie, aussi bien que pour la navigation.

L'avantage principal en utilisant des robots à capacités limitées est leurs coûts ; les capteurs à courte portée sont en général peu coûteux. En raison du coût réduit, il est faisable de construire de grandes équipes avec de tels robots. En plus, les petits robots peu coûteux sont potentiellement jetables - même en grands nombres. La disponibilité est importante dans des applications de sécurité ou de reconnaissance où la nature de l'environnement est telle que les robots ne peuvent pas être récupérés.

Bien que les capacités des capteurs des robots soient limitées, ces capacités sont encore suffisantes pour l'exploration, la navigation et le mapping d'un environnement inconnu. En particulier, ce mémoire se concentre sur la construction des cartes topologiques.

### **1. Cartes topologiques**

Les cartes topologiques, introduit par Kuipers [30], représentent la « connectivité » d'un environnement, généralement présentée sous forme d'un graphe. Les sommets dans les cartes topologiques représentent typiquement « les endroits distincts » dans l'environnement (endroit avec caractéristiques uniques ou les signatures de capteur facilement reconnaissables). Souvent, ces endroits se produisent aux jonctions des couloirs. Les arêtes dans les cartes topologiques peuvent représenter :

- Chemins entre les endroits ;
- Classes des chemins entre les endroits, par exemple tous les différents

chemins qui utilisent un couloir ; ou

- Une tâche ou une séquence de tâches qui déplace le robot d'un endroit à l'autre, et les conditions initiales pour ces comportements.

Un robot avec des capteurs de portée limitée est tout à fait capable d'exécuter des micro-tâches telles que « suivre-mur » et « suivre-couloir ». Ce mémoire se concentre sur la création des cartes topologiques définie en termes de tels micro-tâches.

Ces cartes sont représentées par des graphes dans lesquels les arêtes correspondent aux ordres des micro-tâches qui déplacent le robot d'un endroit à l'autre, et les sommets correspondent aux points de fin des tâches (terminaison).

Les cartes créées de cette façon sont certainement utiles pour la navigation des robots. De plus, elles sont également appropriées pour l'usage humain - en particulier quand quelques informations métriques de l'environnement sont codées. Nos cartes stockent les informations sur les endroits des sommets de carte et les longueurs des chemins entre eux, et ça comme déterminé par l'odométrie du robot.

Cette information odométrique est non précise en raison de l'erreur - provoquée par le glissement des roues et l'inexactitude des capteurs - et ainsi les emplacements exacts des endroits dans la carte ne peuvent pas être connus. Cependant, avec un modèle précis et raisonnable de l'erreur commise par le robot, des estimations probabilistes des mesures métriques prises, peuvent être maintenues, et ces estimations peuvent être améliorées en prenant de nouvelles mesures selon les besoins. En plus, les techniques statistiques peuvent être employées pour comparer les nouvelles mesures prises aux données dans nos cartes (utiles dans la localisation, par exemple).

## **CHAPITRE 2**

# **Les travaux en cartographie et en fusion des cartes.**

Ces dernières années, il y a eu intérêt significatif pour de divers aspects de la construction des cartes en robotique (mapping), de la cartographie pour un simple-robot à la cartographie fortement distribué en multi-robot, et un certain nombre d'algorithmes et de représentations de carte ont été conçus pour résoudre ces problèmes. Ce mémoire repose sur les recherches précédentes en mapping, et s'appuie sur des problèmes relatifs dans d'autres secteurs, tels que l'inférence statistique et le graph matching.

## **2.1 La cartographie d'un robot simple**

Le problème de mapping pour un simple robot a été recherché pendant plus de vingt années. Dans ce temps, deux paradigmes de la cartographie dominants se sont développés : cartographie métrique et cartographie topologique. Dans les cartes métriques, la géométrie du monde est explicitement représentée, par les représentations exactes ou approximatives. Dans les cartes topologiques, des « endroits » dans le monde (typiquement jonctions des couloirs) sont représentés par des sommets dans un graphe, et les chemins entre ces endroits sont représentés par les arêtes. Les cartes métriques fournissent une représentation détaillée du monde mais exigent plus de stockage et sont sensibles aux erreurs de mesure. Les cartes topologiques offrent une représentation plus concise mais, comme traditionnellement mis en application, ne peuvent pas représenter les espaces ouverts. La plupart des techniques de cartographie (métrique ou topologique), sont des techniques probabilistes dans un certain sens. Une récente étude de Thrun [61] Offre une vue d'ensemble sur plusieurs techniques de cartographie probabilistes bien connues.

### **2.1.1 Les cartes de grille d'occupation**

En 1985, Moravec et Elfes [40] ont introduit l'idée des cartes de grille d'occupation pour représenter les informations métriques de l'environnement. Les grilles d'occupation représentent un environnement dans une grille de « cellules » - tout comme les Pixel d'une image numérique. Pour chaque cellule, la probabilité que la cellule est occupée par un certain objet est maintenue. Des travaux plus récents comme dans Konolige [28], Thrun, Fox, et Burgard [59] et d'autres [35, 46] ont raffiné l'approche de grille d'occupation, et ils l'ont appliquée aux problèmes tels que la localisation et mapping simultanés (SLAM).

La plupart des cartes de grille d'occupation sont mises à jour en utilisant des méthodes bayésiennes et ceci quand de nouvelles données sont obtenues. Pagac, Nebot, et Durrant-Whyte [42] ont proposé l'utilisation des fonctions de croyance (Belief)

au lieu des probabilités. Ils emploient la théorie de Dempster-Shafer d'évidence pour maintenir la croyance au sujet de la présence des obstacles en cellules de la grille d'occupation.

### **2.1.2 Les cartes topologiques**

Les cartes topologiques, introduites par Kuipers [30, 31, 43], adoptent une approche différente - extraction des informations sur la connectivité de l'environnement, et dans beaucoup de cas la négligence de la plupart d'information métrique. Certains travaux s'ont concentrés sur l'extraction des cartes topologiques à partir des données des cartes de grille d'occupation [7.58]. Cependant, la plupart des recherches en cartographie topologique, essaient d'acquérir des informations topologiques sur l'environnement pendant l'exploration [33, 50]. Parmi les travaux récents dans ce secteur, Rybski et al.[45] construisent les cartes topologiques en identifiant les caractéristiques visuelles dans les données d'image acquise pendant l'exploration, et Tovar, LaValle, et Murrieta [64] maintiennent une représentation d'un arbre de visibilité mise à jour dynamiquement qui capture la topologie de l'environnement.

Quelques chercheurs ont combiné les deux approches de cartographie métriques et topologiques en maintenant l'information métrique locale (c.-à-d. données d'occupation) aux nœuds topologiques [15, 60]. Tomatis, Nourbakhsh, et Siegwart [62, 63] ont utilisé une carte topologique pour représenter un réseau de couloirs et les cartes métriques pour représenter les salles. En revanche, le travail présenté dans ce mémoire prolonge le paradigme de cartographie topologique pour représenter les espaces ouverts. L'utilisation du comportement « suivre-mur » pour produire la carte initiale d'un terrain, est liée à l'idée « de la navigation côtière » [44] qui reconnaît que les secteurs près des murs et des obstacles ont « un volume élevé d'information ». Une fois que le robot a construit la carte des limites de son espace, il peut sans risque explorer l'intérieur.

### **2.1.3 Fermeture de boucle**

Un des problèmes fondamentaux dans la cartographie topologique est « la fermeture de la boucle ». C'est le problème d'identifier si le robot est revenu à un endroit qu'il a précédemment visité. (Un problème analogue pour les cartes métriques est « le problème de correspondance »).

Une approche simple mais non pratique est « de laisser tomber un caillou » [1] - lors du retour au caillou, la boucle a été fermée. Kuipers et Beeson [32] adoptent une approche différente : ils emploient l'apprentissage supervisé pour identifier « les endroits distinctifs » qui sont les nœuds dans leurs cartes topologiques. Avec un système de détection suffisamment riche, des endroits distincts sont facilement différenciés en enregistrant une « signature » unique pour chaque endroit.

Sans un tel system pour identifier des endroits distincts, le robot doit continuer à traverser l'environnement pour déduire si une boucle a été fermée.

Le « reversal procedure » de Kuipers' [33] encapsule l'idée générale d'utiliser la carte topologique lui même pour prendre cette décision. Choset et Nagatani [8], qui utilisaient des cartes topologiques basées sur le diagramme de Voronoi généralisé de l'environnement, décrivent une approche où les caractéristiques structurelles de la carte (par exemple, le degré de sommets et l'ordre des arcs adjacents) sont les critères primaires pour la vérification.

Tomatis et al. [63] incluent cette comparaison dans un POMDP qui devrait montrer un seul maximum quand la boucle a été fermée.

### **2.1.4 Robots limités**

D'autres ont étudié le problème de cartographie et d'exploration sur les robots avec un système de détection limitée. Butler, le Rizzi, et le Hollis [6] décrivent une méthode pour le problème de la couverture d'un environnement rectilinéaire en utilisant des robots avec des capteurs de contact mais avec une odométrie presque

parfaite. Le problème de la couverture est en grande partie équivalent à la cartographie métrique.

Doty et Seed [13] ont donné des résultats préliminaires dans la création des cartes de « landmark » en utilisant un robot avec quatre capteurs infrarouges à courte portée et un capteur à longue portée de type SONAR. Ils ont adopté une approche semblable au nôtre dans la construction des cartes de base : le robot contourne les obstacles en utilisant des comportements simples « tâche », créant la « cartes des contours » des obstacles. Cependant, leur approche au problème de fermeture de la boucle est quelque peu simple : le robot déduit qu'un obstacle est contourné complètement quand l'angle total est dans l'intervalle  $360^{\circ} \pm 45^{\circ}$  ; la vérification de cette déduction est effectuée en faisant « une deuxième passe » autour de l'obstacle. En plus, ils n'abordent pas les erreurs provenant de l'odométrie ou des capteurs.

### **2.1.5 Les autres travaux**

Notre approche au problème de la fermeture de boucle est basée sur l'accumulation des preuves pour vérifier ou rejeter les hypothèses sur une fermeture de boucle probable. Les preuves sont basées sur des fonctions de croyance dérivées du modèle d'erreur de l'odométrie du robot et ils sont combinées en utilisant la théorie de Dempster-Shafer [41, 47-49]. Cette approche est liée à celle de Cox et de Leonard [10], qui maintient des hypothèses multiples au sujet de l'état d'un monde dynamique. Dans leur travail, la probabilité de chaque hypothèse est initialisée et mise à jour en utilisant une plateforme bayésienne. L'avantage principal d'une méthode basée sur le travail de Dempster-Shafer est la capacité de représenter de « l'ignorance » - le manque de croyance dans l'hypothèse de fermeture de boucle (Hypothèse correcte ou non).

Afin de faire des hypothèses au sujet des boucles dans l'environnement, on se base sur les techniques « de la cartographie stochastique », qui sont été d'abord présentées par Smith et Cheeseman [52] et plus tard ont été détaillés par Leonard et Feder [18.34]. Dans la cartographie stochastique, les endroits sont associés avec des distributions de probabilité des positions qui sont basées sur le modèle d'erreur du

robot. Nos robots maintiennent de telles distributions et les emploient pour produire des hypothèses de fermeture de boucle et pour grouper les preuves sur ces hypothèses.

## **2.2 La cartographie multi-robot et la fusion des cartes**

Intuitivement, une équipe de robots devrait pouvoir créer des cartes d'une manière plus efficace et exacte qu'un simple robot. La recherche récente sur la cartographie coopérative a prouvé que généralement c'est le cas.

### **2.2.1 Les approches**

La plupart du travail sur la cartographie en multi-robot s'est concentré sur la création des cartes de grille d'occupation [22,27,57]. Grabowski et al. [22] ont discutés l'utilisation d'une équipe hiérarchique de robots au problème de l'exploration et du mapping ; dans le niveau le plus bas de leur hiérarchie sont les « millibots », des petits robots avec des capacités de détection minimales.

Un certain travail a été également effectué sur la cartographie « feature-based » ou topologique multi-robot [11, 19, 55]. Jennings, Kirkwood-Watts, et Tanis [26] ont utilisé des robots pour créer des cartes à partir des chemins de Voronoi, et puis, ils ont employé tout simplement la distance entre ces robots pour fusionner ces cartes. Dudek et al. [16] créent des cartes qui se semble à un graphe à partir de l'environnement, ceci est sous la condition que tous les robots commencent au même sommet dans le graphe.

Une grande partie des travaux sur la construction et la fusion des carte multi-robot, suppose que tous les robots dans l'équipe commencent le processus de mapping par une référence commune - une supposition restrictive qui n'est pas faite dans ce mémoire. Une exception à noter, est l'approche adoptée par Ko et al. [27] dans lequel les robots échangent des cartes d'occupation et essaient de se localiser dans chaque carte en utilisant une version de filtre de particules.

Le travail le plus proche de notre sujet est celui de Dedeoglu et de Sukhatme [11], qui ont présenté une méthode pour fusionner des cartes basées sur « les repères »

(landmark) sans référence commune. Ils essayent d'estimer une transformation entre deux cartes en utilisant la correspondance de sommet. L'approche présentée ici est similaire parce qu'elle estime des transformations géométriques entre les deux cartes. Cependant, au lieu d'employer ces transformations pour produire des correspondances de sommet, notre méthode calcule les transformations en utilisant les correspondances qui sont trouvées de la structure des cartes. Cette utilisation de la topologie des cartes rend la découverte des coïncidences potentielles rapide et efficace.

### **2.2.2 Le graphe d'appariement**

Le problème de fusion des cartes topologiques peut être vu dans le cadre des graphes de correspondance. Le graphe de correspondance [9] est un domaine qui a eu de nombreuses recherches pendant les dernières trois décennies, en particulier pour les « Pattern Recognition ».

Quand aucune information métrique n'est incluse dans les cartes, le problème de correspondance est en grande partie structurel de nature - la plupart d'information utile vient de la connectivité des arêtes de la carte - bien qu'il y ait souvent quelques informations disponibles supplémentaires, telles que les types de nœud. En particulier, le problème de fusion des cartes peut être exprimée comme instance du problème Du plus grand sous-graphe commun [4, 17, 38], puisqu'il est souhaitable de trouver la meilleure correspondance entre deux cartes.

### **2.2.3 Le recalage**

Si les cartes maintiennent des informations métriques - par exemple, les endroits estimés, ou la longueur des chemins entre eux - le problème peut être vu comme un problème de l'« image registration and mosaicking ». Une étude complète des différentes techniques appliquées pour le recalage a été faite par Brown en 1992 [3].

La plupart des algorithmes du recalage, tels que l'algorithme itératif très connu (ICP) Iterative Closest Point [2], sont (les algorithmes) itératifs par nature [51, 53, 54].

Les algorithmes itératifs essaient de trouver les transformations entre les frames de l'image en minimisant les distances ceci d'une manière itératives.

L'approche utilisée dans ce mémoire se base sur le graphe d'appariement et le recalage. Les appariements « sont développées » en se basant sur la topologie du graphe de la carte. Ces appariements sont principalement structuraux en nature, bien qu'ils tiennent compte de la similarité des attributs métriques appropriés au sommet et des arêtes. En utilisant ces appariements structuraux, des procédures de la littérature de recalage sont utilisées pour trouver la transformation géométrique qui réduise au minimum la distance entre les sommets appariés. Le groupement des appariements par la similitude de leur transformation les rend des groupes denses. Cette méthode se comporte bien même lorsqu'il y a des régions de chevauchement multiples dans les cartes séparées.

## **CHAPITRE 3**

# **CARTOGRAPHIE AVEC DES ROBOTS LIMITES**

Considérons la situation suivante : on suppose que quelqu'un est entrain de se déplacer dans sa maison dans l'obscurité. Il quitte le lit, mais n'allume pas les lumières (il ne veut pas déranger les autres en maison). Donc il ne peut rien voir, ainsi il cherche par sa main le mur. Quand il le trouve, il continue à se déplacer à l'aide de sa référence (le mur) tout au long du couloir. Il y a de petites discontinuités dans le mur, comme les portes, mais ceci ne va pas arrêter notre bon homme. Il va continuer tout droit. Il sent l'extrémité du mur, ainsi il tourne par le coin et continue sa marche. Peut-être à un certain point, il laisse le mur et il se déplace à l'intérieur d'une pièce (ou bien pour éviter

un obstacle). Ainsi il peut atteindre la cuisine sans beaucoup d'ennui.

Cette navigation « aveugle » est une analogie convenable pour la façon dont un robot a des capacités de détections limitées « voit » le monde. Par ces « moyens de détection limitée », nous voulons dire que le robot a des informations très limitées sur le monde. En raison de ses moyens de détection limités, la vue du robot de son environnement est sévèrement restreinte- les différentes informations du monde peuvent seulement être détectés à une courte distance et en observant une séquence chronologique de lectures des capteurs.

Afin de créer des cartes avec des équipes de robots coopérants, les problèmes de l'exploration, la cartographie et la navigation avec un robot simple doivent d'abord être abordés. Notre problème est de faire laisser un simple robot explorer un environnement inconnu, qu'il crée une carte topologique qu'il peut plus tard l'employer pour se diriger entre deux endroits quelconques déjà visités. Les chemins trouvés doivent être efficaces par rapport aux moyens de détection limitée, les erreurs de détection et de l'odométrie.

Ce chapitre démontre que les robots à des capacités de détection limitées peuvent accomplir cette tâche, bien qu'il y ait quelques limitations, en particulier sur l'exploration complète de la carte, suivant la taille et la complexité géométrique de l'environnement. Nous ne devrions pas s'attendre à la même performance d'un robot avec des possibilités de détection plus riches (par exemple les capteurs lasers). Tandis que nous permettrons peu de performance en termes de temps d'exécution et portée, nous attendons toujours à une exactitude raisonnable.

Notre approche construit une carte initiale qui peut être considérée comme une carte topologique. Dans cette carte, les sommets sont des endroits où un comportement (micro-tache) de robot se termine, et les arêtes représentent un ordre de comportements qui déplacent le robot d'un endroit à un autre. En utilisant cette philosophie, nous pouvons dépasser la faiblesse traditionnelle des cartes topologiques en représentant les espaces ouverts. D'abord, nous allons présenter le concept des

portails qui lient les espaces ouverts (déterminé par le comportement suivre-mur) aux couloirs (déterminé par le comportement suivre-hall). En suite, nous présentons l'idée des incursions, qui produise des liens directs entre les sommets dans la carte.

Notre utilisation des robots simples nous a également menés à une nouvelle approche pour le problème de « la fermeture de la boucle » - ceci est assuré en déterminant que le robot est revenu à un endroit qu'il a déjà visité. Puisque les robots simples ne peuvent pas identifier un endroit d'une manière unique basé sur les lectures instantanées des capteurs, on suppose que le robot a fermé la boucle en se basant sur l'évaluation géométrique de sa position, puis on essaye de vérifier cette hypothèse.

### **3.1 Préentions**

On considère qu'un robot simple qui a un nombre restreint de capteurs à courte portée ; ces capteurs renvoient la distance  $r$  qui est la distance entre le robot et l'obstacle le plus proche en suivant une ligne droite dans la portée maximum. On note que « la courte portée » est relative aux dimensions de l'environnement. Il doit y avoir assez de capteurs pour l'exécution robuste des comportements décrits dans la section 3.2.1.

Il y a des erreurs connues dans le mouvement du robot, mesures de l'odométrie et les mesures des capteurs. En particulier, la distance parcourue par un robot et l'angle qu'il tourne peuvent être non précis.

Sans aucune procédure de « feed-back », le robot peut ne pas se déplacer en ligne droite. On suppose que ces erreurs sont aléatoires et de moyenne de zéro, mais on ne va pas supposer une distribution de probabilité particulière. On suppose que le robot ne commit aucune erreur dans la détection « des types » des endroits qu'il rencontre.

On suppose que l'environnement est fermé, statique et rectiligne. Cependant la plupart de nos idées se prolongent directement aux environnements polygonaux généraux, la prétention rectiligne enlève l'incertitude dans l'orientation du robot, et ceci



**FIG 3.1 : Suivre-mur et détection des caractéristiques. La zone « sûre »  $[r_{\min}, r_{\max}]$  est indiquée par le secteur en ombre. Des sommets sont placés à une distance de  $r_0$  des deux murs d'incident d'un coin bien défini. Le coin extrême gauche est extérieur ; l'extrême droit est intérieur. Le reste des discontinuités dans le mur constituent les caractéristiques mal définis.**

Le comportement suivre-mur se termine quand le robot détecte un coin-bien-défini. Un coin-bien-défini extérieur se produit quand le mur s'éloigne par plus de  $r_{\max} - r_0$  (c.-à-d., un robot qui suit un mur à  $r_0$ , voit le mur disparaître au delà de  $r_{\max}$ ) ; de même, un coin-bien-défini intérieur se produit quand le mur déborde vers l'intérieur plus que  $r_0 - r_{\min}$ . N'importe quelle discontinuité qui demeure dans la marge  $[r_0 - r_{\min}, r_{\max} - r_0]$  est un coin-mal-défini (en général inutilisé dans nos cartes).

$$r_* = \sqrt{r_0^2 + r_{\min}^2} \quad (3.1)$$

Dans le cas d'un coin extérieur, le comportement suivre-mur se termine à une distance de  $r_0$  après la discontinuité — le mouvement ouvrir-boucle doit être effectué. Pour un coin intérieur, le comportement suivre-mur s'arrête à une distance  $r_0$  avant la discontinuité.

### **SUIVRE-COULOIR**

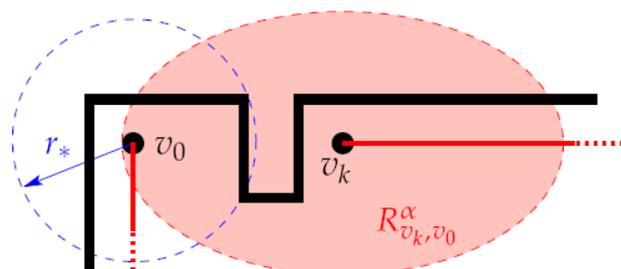
Le comportement SUIVRE-COULOIR est similaire au comportement suivre-mur, sauf qu'il est seulement employé quand le robot traverse un « couloir » d'une largeur  $\leq$

$2r_0$ . Dans ce cas, le robot détecte les deux murs. Ainsi il se déplace par le centre du couloir. Le robot peut rencontrer des « jonctions » des couloirs ou des coins. Pour ces jonctions et coins le robot doit effectuer le mouvement ouvrir-boucle comme dans la navigation des coins extérieurs dans le comportement suivre-mur.

### 3.2.2 Le processus de cartographie

D'abord, on considère un environnement qui se consiste entièrement des espaces « ouvert » - un environnement qui ne contient aucun espace « étroit » de largeur moins que  $2r_0$ . Essentiellement, les espaces ouverts sont des secteurs dans lesquels le robot ne peut pas « voir » des obstacles avec les capteurs de côté simultanément. (La prétention des espaces ouverts est enlevée dans la section 3.4.) Le procédé de base de cartographie est comme suit :

- D'une position initiale arbitraire, le robot lance le comportement DEPLACER-VERS-MUR, qui aligne le robot avec un certain mur dans l'environnement.
- Le comportement de SUIVRE-MUR est activé, et son point de terminaison devient le sommet de début  $v_0$  - le premier sommet dans notre carte topologique.
- Après la découverte de  $v_0$ , le robot se tourne comme il le faut pour suivre le prochain mur (incident à  $v_0$ ). Les sommets sont ajoutés pour chaque caractéristique bien défini de l'environnement et qui termine le comportement SUIVRE-MUR. Chaque sommet est placé à une distance  $r_0$  de chaque mur incident. Chaque nouveau sommet est relié, à une arête non dirigé, au sommet précédemment découvert (c.-à-d.  $v_0 \leftrightarrow v_1$ ,  $v_1 \leftrightarrow v_2$ , et ainsi de suite).
- Ce processus se répète jusqu'à le retour à  $v_0$ . Ceci est garanti de



se produire, puisque l'environnement est fermé.

**FIG 3.2 : Une situation dans laquelle nous présumons que  $v_k = v_0$ . La région  $R^{\alpha}_{v_k, v_0}$  représente l'incertitude de position dans l'endroit  $v_k$  en ce qui concerne  $v_0$ , et est liée selon la limite de fiabilité  $\alpha$ , un paramètre de l'algorithme. Puisque cette région contient  $v_0$ , il est possible que les deux sommets soient identiques. Cependant, puisque la région n'est pas complètement liée par le cercle de rayon  $r^*$  autour de  $v_0$ , nous ne sommes pas certains que l'hypothèse soit correcte, et devons obtenir davantage d'évidence avant de prendre une décision**

La reconnaissance du retour du robot au sommet de début est le problème de la « fermeture de la boucle. »

### 3.3 Fermeture de boucle

Le problème de fermeture de boucle est difficile quand le robot n'a aucune connaissance antérieure de la complexité de l'environnement, et un certain nombre de différentes méthodes pour résoudre le problème a été proposées [1, 8, 32, 33, 63]. Nous adoptons l'approche suivante :

- D'abord, nous devons être capables d'identifier un retour probable au sommet de début  $v_0$ . L'identification de cette possibilité est atteinte en « supposant » (faire une hypothèse) que nous avons fermé la boucle.
- Lors de faire une telle hypothèse, nous devons la vérifier ou la rejeter. Il y a plusieurs méthodes; nous proposons une approche basée sur la théorie de Dempster-Shafer.

#### 3.3.1 La construction d'une hypothèse

Pour chaque sommet  $v_i$  dans la carte topologique, nous employons le modèle d'erreur d'odométrie du robot pour maintenir une fonction de densité de probabilité  $U_{v_i, v_0}$  qui décrit l'endroit de  $v_i$  par rapport à  $v_0$ . L'incertitude dans cette distribution augmente monotoniquement pendant que la carte augmente. Pour un niveau de confiance  $\alpha$  spécifique à notre algorithme de cartographie, laissons  $R_{v_i, v_0}^\alpha$  représente les limites de confiance sur  $U_{v_i, v_0}$ . Ceci est la région dans laquelle nous croyons que le robot s'est localisé avec le paramètre de confiance  $\alpha$ .

Si le robot atteint un sommet  $v_k$  qui est le même type que  $v_0$  (par exemple tous les deux sont des coins intérieurs, ou tous les deux sont des coins extérieurs) et  $v_0$  se trouve en  $R_{v_k, v_0}^\alpha$ , alors, on crée une hypothèse

$$H_0 \equiv v_k = v_0 \quad (3.2)$$

Que le sommet  $v_k$  ferme la boucle. Voir le Fig 3.2 pour un exemple d'une telle situation.

Notons que si la région entière  $R_{v_k, v_0}^\alpha$  se trouve complètement dans un rayon  $r^*$  de  $v_0$  (généralement pas le cas), on peut déclarer immédiatement que l'hypothèse est correcte avec au moins la confiance spécifiée puisqu'aucun autre sommet du même type ne peut être dans ce rayon de  $v_0$ .

### 3.3.2 Vérification d'une hypothèse

Une fois qu'une hypothèse de fermeture-de-boucle est faite, nous essayons de la vérifier. La possibilité pour faire ceci est de faire marche-arrière - voyager dans les deux sens entre le sommet  $v_k$  et le sommet  $v_0$ , en prenant des mesures additionnelles de toute les arêtes. En supposant que les erreurs sont aléatoire et de moyenne zéro, faisant marche-arrière nous permet de réduire l'incertitude de position en fusionnant des mesures multiples des mêmes arêtes. La marche arrière réduira par la suite la taille de  $R_{v_k, v_0}^\alpha$  jusqu'à ce qu'elle se trouve entièrement à l'intérieur de  $r^*$  de  $v_0$  ou jusqu'à ce que  $v_0$  ne se situe pas dans  $R_{v_k, v_0}^\alpha$ . C'est la seule manière sûre de vérifier ou rejeter une

hypothèse, mais elle est non pratique et exige beaucoup de voyages entre le sommet  $v_k$  et le sommet  $v_0$ .

Au lieu de cela, après la construction de l'hypothèse de fermeture de boucle, notre robot continue à traverser des arêtes « vers l'avant », comparant les caractéristiques qu'il rencontre et les mesures prises à celles qu'il estime trouver sous l'hypothèse. Si nous venons sur une certaine différence structurale (par exemple sommets de différents types ou des arêtes incidentes à différentes orientations), alors l'hypothèse est rejetée. Cependant, afin de vérifier l'hypothèse, nous comparons les longueurs des arêtes qui devraient être égales sous l'hypothèse, et on utilise la comparaison pour établir une « preuve » pour ou contre l'hypothèse en utilisant des méthodes basées sur la théorie de Dempster-Shafer de l'évidence [49].

Dempster-Shafer nous donne une manière d'assigner la « croyance » à un ensemble de possibilités - dans ce cas, l'hypothèse est correcte et on note C, ou incorrecte et on note I. Il nous fournit également une manière de représenter l'« ignorance » pour ces possibilités. L'ignorance, dénotée  $\{C, I\}$ , peut être considérée comme l'incertitude que l'une des possibilités est vérifiées par « preuve ». La représentation de l'ignorance est particulièrement très utile dans des situations incertaines comme dans notre situation.

Pour chaque possibilité C, I et  $\{C, I\}$ , nous maintenons une probabilité de base pendant que nous recueillons des preuves :  $m(C)$ ,  $m(I)$  et  $m(\{C, I\})$ , respectivement. En ensemble, ces probabilités de base constituent une fonction de croyance. La théorie de Dempster-Shafer fournit des règles pour combiner des fonctions de croyance construites à partir d'ensembles multiples de preuve.

On suppose que nous avons construit une hypothèse  $H_0 =$  « deux sommets dans notre carte sont identiques ». Puis, on continue notre déplacement vers l'avant dans l'environnement, on prend une série de nouvelles mesures de la longueur de mur. Sous l'hypothèse  $H_0$ , chaque nouvelle mesure  $l$  a une ou plusieurs correspondance, ces correspondance sont des mesures qui ont été prisent précédemment et qui commencent et finissent aux mêmes sommets dans la carte comme la mesure  $l$ .

La mesure  $l$  et les mesures correspondante constituent un ensemble de mesure L:

$$L = \{l_0, l_1, \dots, l_n\} \quad (3.3)$$

avec  $n \geq 2$ . Sous  $H_0$ , toutes les mesures en L proviennent du sommet commun  $v_i$  et finissent au sommet commun  $v_j$ ; l'ensemble L correspond ainsi à une arête simple dans le graphe de la carte. On souhaite déterminer l'évidence donnée pour notre hypothèse par l'ensemble L. On utilise la théorie de Dempster-Shafer pour fusionner l'évidence fournie par les ensembles de telles mesures (chaque ensemble correspond à une paire de sommets adjacentes différente dans la carte). Avec le temps, l'évidence s'accumule pour (ou contre)  $H_0$ .

Il est raisonnable d'utiliser seulement la comparaison de différentes mesures « cote-à-cote » pour déterminer l'évidence – le contraire est de comparer la « géométrie » (forme) de la carte déterminée par les mesures de plusieurs murs - parce que pour une hypothèse correcte, les erreurs dans chaque mesure sont indépendantes. Une grande anomalie parmi un ensemble de mesures précédentes d'un seul mur devrait certainement donner l'évidence contre l'hypothèse.

Donné un ensemble de mesures  $L$  et leurs distributions d'incertitude associées, nous devons déterminer des valeurs de base de probabilité pour  $m_L(C)$ ,  $m_L(I)$  et  $m_L(\{C, I\})$  tels que

$$m_L(C) + m_L(I) + m_L(\{C, I\}) = 1 \quad (3.4)$$

Ici,  $m_L(C)$  représente notre croyance, en supposant que pour les mesures en  $L$ ,  $H_0$  est correct, alors  $m_L(I)$  représente notre croyance que  $H_0$  est incorrect, et  $m_L(\{C, I\})$  représente notre « ignorance » - le degré pour lequel  $L$  ne fournit ni l'évidence pour l'hypothèse ni contre. La fonction par laquelle nous déterminons  $m_L(C)$ ,  $m_L(I)$  et  $m_L(\{C, I\})$  se nomme une « fonction de croyance. »

Notre fonction particulière de croyance est basée sur des méthodes bien connues d'inférence statistique. Elle calcule essentiellement les probabilités que les mesures en  $L$  ont été prises du même mur en prenant en compte le modèle d'erreur du robot.

Comme nous prenons de nouvelles mesures, nous devons combiner l'évidence qu'elles fournissent avec l'évidence précédente que nous avons acquise, afin de mettre à jour notre croyance globale dans l'exactitude de l'hypothèse. Pour combiner l'évidence, nous employons la règle de Dempster de la combinaison. Dans notre application particulière, les règles de combinaison sont comme suit : étant donnée une croyance globale  $m$  et une croyance  $m_L$  attribuée à un ensemble de mesures spécifique  $L$ ,

$$m \oplus m_L(C) = \frac{m(C)m_L(C) + m(C)m_L(\{C, I\}) + m(\{C, I\})m_L(C)}{1 - m(C)m_L(I) - m(I)m_L(C)} \quad (3.5)$$

$$m \oplus m_L(I) = \frac{m(I)m_L(I) + m(I)m_L(\{C, I\}) + m(\{C, I\})m_L(I)}{1 - m(C)m_L(I) - m(I)m_L(C)} \quad (3.6)$$

Nous continuons de construire l'évidence sur l'hypothèse de fermeture de boucle tant que notre croyance globale en l'exactitude (ou l'inexactitude) de l'hypothèse n'a pas dépassé un certain seuil  $\beta$ . Généralement on choisit  $\beta = \alpha$ .  $\alpha$  est le niveau de confiance que nous avons utilisé en construisant l'hypothèse.

### 3.3.3 Discussion

Les mondes en lesquels il est plus difficile de fermer la boucle sont ceux avec des « similarités structurales ». Deux manières dont un monde pourrait avoir des « similarités structurales » :

- Se compose de séquences similaires de murs que « se développent en spirales » vers l'intérieur ou à l'extérieur ;
- Consiste entièrement ou partiellement en « des répétitions de séquences semblables » de murs.

Notre approche présentée va bien en mondes « en spirale », tant que l'erreur éprouvée par le robot n'est pas grande relativement à l'« étroitesse » de la spirale. Les mondes de « séquences similaires répétées » sont généralement plus difficiles, bien qu'ils puissent souvent être correctement tracés malgré la présence de l'erreur, en utilisant un seuil assez suffisant pour l'acceptation d'une hypothèse.

La continuation de la navigation de l'environnement en avant après la construction d'une hypothèse sur la fermeture de boucle présente un certain nombre de souci. Le souci principal avec cette approche est qu'on ne peut jamais être complètement sûr qu'une hypothèse est correcte ; c'est une insuffisance de presque

toutes les méthodes basées sur des hypothèses pour fermer des boucles. L'approche de marche arrière résout ce problème, mais seulement après un nombre indéfini de mesures. Nous croyons que l'idée de construire l'« évidence » pour ou contre l'hypothèse est intuitive et raisonnable pour l'usage dans des situations pratiques.

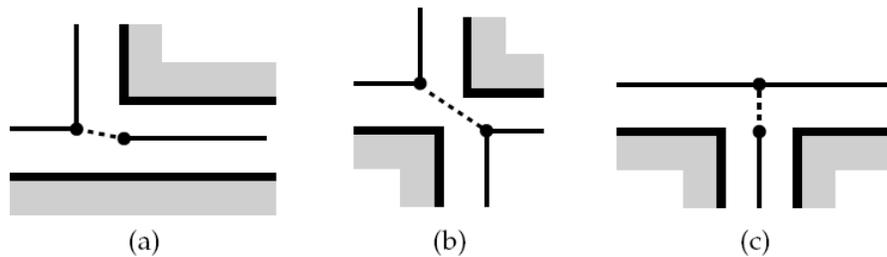
Nous avons fait plusieurs autres observations sur le problème de fermeture de boucle :

- En retournant vraiment à  $v_0$ , le robot aura réfuté toutes les hypothèses contradictoires par la non correspondance structurales/topologiques. Cependant, le robot ne peut pas savoir quand il est revenu à  $v_0$  parce que la complexité du monde est inconnue.
- L'orientation du robot doit passer  $360^\circ$  avant qu'il puisse retourner au sommet de début.
- Les fermetures de boucle fausse positivement ou négativement ne sont pas des « problèmes d'arrêt » pour notre approche (nous pouvons les réfuter plus tard), mais il est préférable de trouver la bonne hypothèse pendant le processus initial de la cartographie. Ainsi, les seuils d'acceptation devraient être raisonnablement hauts à moins que la connaissance antérieure de l'environnement indique que le problème de similarité ne sera pas un souci - souvent le cas.
- Dans les scénarios de cartographies topologiques générales dans lesquels les sommets peuvent avoir un degré plus que deux, nous devrions choisir le chemin à une jonction que nous comptons rapporter la plupart d'évidence.
- La connaissance antérieure de quelques aspects du monde (par exemple distributions potentielles des longueurs des murs.) pourrait mener à une fermeture de boucle plus efficace en nous permettant de concevoir une fonction de croyance et en nous laissant être plus décisif en acceptant ou en rejetant des hypothèses.

### 3.4 Portails et couloirs étroits

Jusqu'ici, nous avons seulement adressé le cas où l'espace ouvert se trouve à côté du robot. Quand la région autour du robot est assez étroite, le robot peut voir les murs des deux côtés, le robot devrait commuter de suivre-mur au comportement suivre-couloir. Nous définissons des portails pour établir le rapport entre ces deux modes différents du processus de la cartographie.

Nous pouvons prolonger notre processus de cartographie de base pour prendre en considération les portails afin de pouvoir traverser des couloirs étroits qui peuvent nous mener à d'autres espaces ouverts, laissant ces régions pour une future exploration. Ceci a l'avantage de rendre les boucles dans la carte de base plus petites et donc plus facile à se fermer.



**FIG 3.3 : Les portails se produisent dans des conditions de terminaison suivre-mur ou suivre-couloir spéciales. Chaque figure, le robot se déplace de gauche à droite. Dans la figure (a), le portail du coin intérieur est détecté tout en suivant un mur. Dans la figure (b), le portail, détecté tout en tournant un coin intérieur, mène directement à un autre espace ouvert. Dans la figure (c), le portail est détecté tout en tournant un coin extérieur.**

On considère un couloir étroit avec une largeur  $\leq 2r_0$ , et on suppose que le robot peut faire seulement des suivre-couloir dans des couloirs qui sont au moins de  $2r_w$ . Un portail se produit à un endroit où ces conditions apparaissent d'abord tout en suivant-mur, ou où elles disparaissent après exécution d'un suivre-couloir (pour des exemples, voir Fig 3.3).

Dans nos cartes, les portails sont généralement représentés par des arêtes spéciales qui sont connectées aux sommets résultants des suivre-couloir.

### **3.5 La navigation en utilisant la carte de base**

Après que nous ayons complété le processus de la cartographie de base, la navigation entre deux points quelconques dans la carte est directe : nous employons une recherche dans le graphe de la carte pour construire un ordre de comportements, et puis on exécute ces comportements. Cependant, la navigation par la carte de base est inefficace dans beaucoup de cas : le robot peut devoir naviguer la moitié d'un bâtiment pour atteindre un point à travers un grand hall. La deuxième phase de notre algorithme présente des améliorations dans la carte qui améliorent son utilité pour se diriger par les espaces ouverts (et pour découvrir des « îles » dans ces espaces si elles existent).

### **3.6 Améliorations**

Les améliorations à notre carte de base tentent de relier des endroits disjoints dans l'environnement avec les chemins qui traversent des régions (sans particularité) ouvertes. Une insuffisance commune de beaucoup de méthodes de cartographie topologiques est l'incapacité de traiter le cas des « espaces ouverts ». Les améliorations offrent un outil pour prolonger les cartes topologiques à de tels espaces d'une manière normale. On note que les améliorations s'appliquent seulement aux parties produite par le comportement suivre-mur ; les améliorations sont inutiles dans les couloirs étroits explorés par suivre-couloir.

Nos améliorations sont basées sur la géométrie de la carte de base. Ainsi, avant de produire des améliorations possibles, nous devons inclure les sommets de la carte dans le plan. En raison de l'erreur de mesure pendant la cartographie de base, une fois que nous fermons la boucle, la carte n'est plus planaire - nos mesures sont géométriquement contradictoires avec l'hypothèse de fermeture-boucle que nous avons acceptée.

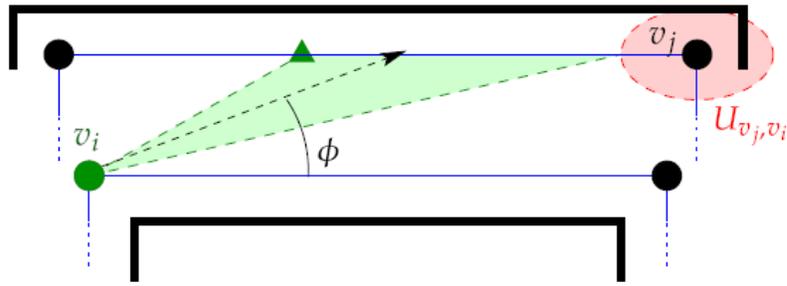
Le problème de construire une carte géométrique cohérente à partir des mesures de la distance locale et angulaires dans une carte topologique a été abordé par plusieurs chercheurs, y compris Duckett et al. [14], Lu et Milios [36], et Golfarelli et al [21]. Une seule méthode suffit.

### **3.6.1 Les incursions**

On suppose qu'on veut ajouter un chemin reliant le sommet  $v_i$  avec le sommet  $v_j$  dans notre carte, où  $v_i$  et  $v_j$  ne sont pas déjà reliés dans la carte de base. Si notre robot n'éprouvait aucune erreur de mouvement, nous pourrions faire ceci simplement en lançant un comportement pour lequel le robot (à  $v_i$ ) tourne vers  $v_j$  et se déplace directement jusqu'à ce qu'il atteigne sa cible. Malheureusement, nous devons tenir compte de l'erreur, dans notre carte de base et dans l'« exploration » de l'espace ouvert.

Nous avons déjà des moyens pour estimer l'incertitude relative à des endroits de sommet, en utilisant des distributions d'incertitude basées sur le modèle d'erreur du robot et centrées à chaque sommet. Les limites de fiabilité spécifient une région dans laquelle nous croyons (avec la confiance désirée) que le sommet se trouve dedans.

Nous supposons que la « diffusion » de l'erreur d'évaluation de l'endroit du robot, pendant qu'il explore  $v_i$ , peut être liée par une zone angulaire limitée commençant à  $v_i$ . L'incertitude de position du robot se développe avec la longueur de l'exploration.



**FIG 3.4 : Une amélioration du sommet  $v_i$  au sommet  $v_j$ . Le robot se tourne vers le mur opposé et « fait une incursion » d'angle  $\phi$ . L'incertitude angulaire du robot (accentuée) est assez petite qu'elle est confiante il débarquera sur le mur cible.**

Il y a deux chemins potentiels de n'importe quel sommet  $v_i$  à un autre sommet  $v_j$ , qui correspondent aux deux murs incidents au  $v_j$ . Pour chaque chemin, le robot fait une exploration de telle manière qu'il tombe sur le mur désiré. Ainsi, une amélioration valide est possible seulement quand l'incertitude de position accumulée par le robot pendant une exploration est moins que toute la longueur du mur cible (entre les limites de confiance dans les endroits de ses points finaux). Pour un exemple d'exploration, voir Fig 3.4.

Les améliorations sont représentées comme des arêtes entre les sommets dans la carte. On va associer à ces arêtes la « séquence de comportement » suivante :

1. Tourner vers le mur cible (incident au sommet cible) avec un angle  $\phi$ ;
2. DEPLACER-VERS-MUR;
3. Exécuter un SUIVRE-MUR en direction du sommet cible.

Nous pouvons faire deux types d'amélioration avec cette approche : améliorations passives, qui peuvent être calculées directement de la carte de base, et améliorations actives, qui exigent une future exploration de l'environnement.

### **3.6.2 Améliorations passives**

L'amélioration passive utilise la connaissance sur la région « balayée » par les capteurs du robot comme il a effectué son exploration initiale. L'union de toutes les régions balayées par les capteurs du robot est « l'espace connu. » Une approximation simple de l'espace connu peut être obtenue en stockant la plus courte portée détectée par la capteur opposée de la capteur suivre-mur en passant par chaque mur.

En utilisant la géométrie de notre carte de base, nous pouvons immédiatement déterminer quel améliorations passent seulement par l'espace connu, et ajoutons ces améliorations à notre carte.

### **3.6.3 Améliorations actives / Exploration**

Les améliorations potentielles qui passent partiellement ou entièrement par l'espace inconnu constituent les « cibles de l'exploration ». Nous devons activement explorer les chemins suggérés par ces améliorations afin de vérifier qu'elles peuvent être traversées ; il peut y avoir des obstacles ou des « îles » dans l'espace inconnu.

Supposant que, en faisant une exploration, le robot rencontre un obstacle. Dans ce cas le robot doit pouvoir retourner sans risque à un endroit connu dans sa carte de base - sinon, il est « perdu » et devra reconstruire sa carte de base ! (Alternativement, le robot pourrait employer des techniques de filtrage par motif, ou une approche basée sur les preuves, afin d'essayer (le robot) de se localiser quand il pense qu'il est revenu à la carte de base.) Généralement avant de faire des explorations, nous jetons ceux qui sont si longue et que le robot peut être non capable de retourner à un endroit connu en raison de l'accumulation de l'incertitude.

On calcule un ordre dans lequel on explore les cibles restantes, par exemple on utilise une approche du voyageur de commerce. Nous visitons alors chaque cible, et on exécute l'ordre de comportement associé.

Si nous ne rencontrons aucun obstacle jusqu'à ce que nous atteignons notre mur cible, nous gardons l'amélioration ; autrement nous la jetons. Selon le modèle de l'erreur du robot, on peut avoir une marge de sécurité pour explorer un obstacle que nous rencontrons ; si oui, on crée une carte de base de l'obstacle, et on ajoute les améliorations en reliant la nouvelle carte à notre carte originale.

### **3.7 Limites**

Notre approche de cartographie n'est pas sans limitations. En particulier, nos robots sont limités dans les mondes qu'ils peuvent construire leurs cartes avec succès. Ces limitations dépendent principalement des insuffisances de détection du robot ; par exemple, nous ne devrions pas nous attendre à ce qu'un robot avec une grande erreur d'odométrie soit capable de construire une carte complète d'un espace vaste et ouvert d'une manière efficace. Tandis que la création des cartes précises des grands espaces avec un robot à des capacités limitées est certainement possible, notre approche tend vers l'efficacité de la cartographie et de la navigation plutôt que la perfection.

Nos robots ne capturent pas chaque caractéristique de l'environnement. En fait, « on adoucis » les détails de petites caractéristiques explicitement en faveur de capturer la structure globale de l'environnement. Ceci diminue le risque de capturer les inexactitudes, qui se produisent plus fréquemment avec les petites caractéristiques.

Le problème de ne pas pouvoir détecter de petites caractéristiques en raison de la basse densité de capteur - est un autre souci de la création des cartes avec des robots limités. Il y a des situations dans lesquelles la faible densité des capteurs du robot permet au robot de rencontrer un obstacle sans le détecter. Par exemple, le robot peut tomber sur un pied d'une chaise. Pour de telles situations, les capteurs de type « bump » peuvent être nécessaires afin de créer des cartes précises. Les capteurs de type « bump » peuvent être considérées en tant que capteurs à courte portée, et en conséquence peuvent être incorporées à l'algorithme de cartographie.

## CHAPITRE 4

# FUSION DES CARTES TOPOLOGIQUES

On suppose qu'on a libéré deux robots sans référence commune dans un environnement et on leur permet de construire des cartes partielles du monde. On veut construire une carte simple et unifiée à partir des deux cartes des deux robots par la superposition des deux régions découvertes. La notion de fusionner des cartes séparées d'une manière efficace est importante pour les systèmes multi-robot qui doivent construire de grandes cartes des environnements inconnus d'une manière exacte et rapide.

Rappelons que les cartes topologiques emploient un graphe pour représenter les différentes possibilités pour la navigation d'un endroit à un autre dans un environnement ; les sommets représentent certains « endroits » dans l'environnement et les arêtes représentent les chemins (ou des classes des chemins) entre ces endroits. Souvent les sommets et les arêtes sont annotés avec des informations métrique, telle que la longueur du trajet (pour des arêtes) ou les orientations relatives des chemins incidents (pour des sommets). Typiquement, les sommets sont des jonctions de deux couloirs droits ou plus, et les arêtes représentent un chemin dans un couloir d'une jonction à une autre. Les cartes topologiques fournissent une description concise de l'environnement et sont spécifiquement adapté à la navigation.

Les robots qui ont exploré des régions d'un environnement qui se chevauche, devraient avoir des cartes topologiques qui ont des sous-graphes communs avec une structure similaire. La solution du problème fusion des cartes est analogue à identifier un appariement entre les sous-graphes dans les deux cartes. Généralement, les attributs des sommets exactement connus, tels que le degré des sommets, correspondent

parfaitement. Cependant, les attributs des sommets et des arêtes, tel que la longueur d'une arête ou l'orientation des arêtes partant d'un sommet, doivent seulement correspondre étroitement.

Si la carte topologique d'un robot contient des mesures de longueur des arêtes et les orientations des arêtes incidentes aux sommets, il y a assez d'information pour calculer des endroits prévus pour un sommet. Ceci suggère que le problème de fusion des cartes pourrait être résolu en utilisant les techniques de recalage (En traitement d'image). Les algorithmes les plus employés couramment pour le recalage sont les algorithmes ICP (Iterative Closest Point). Une correspondance initiale entre les différents points doit être fournie ; l'algorithme estime d'abord une transformation entre les deux ensembles de caractéristiques en réduisant au minimum l'erreur entre les caractéristiques correspondantes. Après, la correspondance est augmentée en cherchant les caractéristiques qui sont présentes et la transformation ré-estimée. Ce processus continue jusqu'à ce que le changement de l'évaluation de transformation entre les itérations soit petit.

Dans ce chapitre, nous décrivons un algorithme qui emploie l'aspect structural d'une correspondance des sous-graphes et l'aspect géométrique du recalage. On crée d'abord, une correspondance en groupant les sommets compatibles entre eux dans les deux cartes, puis on augmente notre correspondance par l'ajout des sommets en utilisant seulement la structure du graphe et les annotations des sommets et des arêtes. On peut éliminer beaucoup d'hypothèses dans cette phase en raison des incompatibilités dans la structure des cartes, ou en raison des annotations incompatibles (telles que les longueurs de arêtes qui sont trop différentes). Après l'augmentation, nous estimons les transformations géométriques pour chaque hypothèse et exécutons le groupement dans l'espace de transformation. Le résultat de notre algorithme est une liste numérotée de groupe, où chaque groupe est un ensemble d'hypothèses pour lesquelles nous pouvons estimer une simple transformation qui a une erreur acceptable entre tous les sommets dans le groupe.

## **4.1 Suppositions**

Les cartes à fusionner doivent être conformes : aucun deux sommets ne peuvent représenter le même « endroit. » Ceci signifie que les robots peuvent identifier s'ils ont revisité un endroit et sont ainsi capable de « fermer la boucle. » (Pour plus de détail sur l'uniformité de carte et le problème de fermeture de boucle, voir le chapitre précédent. On note que notre algorithme de fusion de carte peut être appliqué avec les cartes topologiques d'une manière générales et non pas seulement les cartes créées par l'algorithme décrit plus tôt.)

Les robots doivent enregistrer assez d'information. La connaissance des formes et des longueurs de chemin et les orientations des chemins partant de chaque sommet est suffisante. Nous supposons qu'il y a un modèle d'erreur connu pour les mesures, et que seulement une translation et une rotation (mais aucune mise en échelle) est nécessaire pour fusionner les cartes.

Bien que la plupart de nos exemples soient des mondes rectilignes, l'algorithme que nous développons ici fonctionnera avec toutes les cartes topologiques bidimensionnelles. D'une façon générale, le calcul additionnel est exigé pour les mondes non-rectilignes en raison de la rotation sans restriction.

## **4.2 Construction de l'hypothèse**

La première phase de notre algorithme est de créer des hypothèses en augmentant une correspondance d'un seul sommet. Une hypothèse est une liste de correspondances des sommets et des arêtes entre deux cartes ; pour retrouver ces correspondances on cherche essentiellement tous les sous-graphes communs entre les deux cartes.

### **4.2.1 Correspondances des sommets**

Nous commençons par une paire de sommets compatibles, un de chaque carte. Deux sommets sont évalués pour la compatibilité en examinant leurs attributs : les

attributs exactement connus (tels que le type de sommet) doivent s'égaliser parfaitement, mais des attributs inexact connus (tels que les orientations ou les longueurs des arêtes à un sommet) doivent être comparés en utilisant une fonction de similarité.

Il se semble qu'il est raisonnable de supposer que les robots puissent savoir exactement le degré d'un sommet ; généralement les robots devraient facilement pouvoir déterminer le nombre de chemins partant d'un sommet. En mondes dynamiques, où le degré de sommets peut se changer- due à la fermeture ou l'ouverture des portes - le degré de sommet ne peut pas être traité comme attribut exact.

Les orientations des arêtes relatives à un sommet sont typiquement connues avec une certaine incertitude, ainsi elles doivent être comparées en utilisant une fonction de similarité. L'une des méthodes est de trouver la rotation entre les deux sommets qui réduit au minimum l'erreur entre les orientations des arêtes correspondantes. La quantité d'erreur relative au degré de sommet, peut être employée comme mesure de similarité. Nous avons placé un seuil sur le niveau de similarité requise afin de décider le groupement de deux sommets ou non.

À la conclusion de cette étape, nous aurons beaucoup d'appariement, chacun se compose d'une seule paire de sommets et de la correspondance entre les arêtes. On note qu'une seule paire de sommets peut être conforme sous plusieurs différentes correspondances d'orientation d'arête ; chacune des paires est présente comme un appariement séparée.

#### **4.2.2 Augmentation des appariements**

Maintenant on augmente chaque appariement en examinant les paires d'arête correspondantes partant de l'appariement. Si les arêtes sont compatibles et les sommets aux extrémités sont également compatibles, alors on les ajoute à l'appariement. Si les arêtes ou les sommets sont incompatibles, l'appariement est rejeté

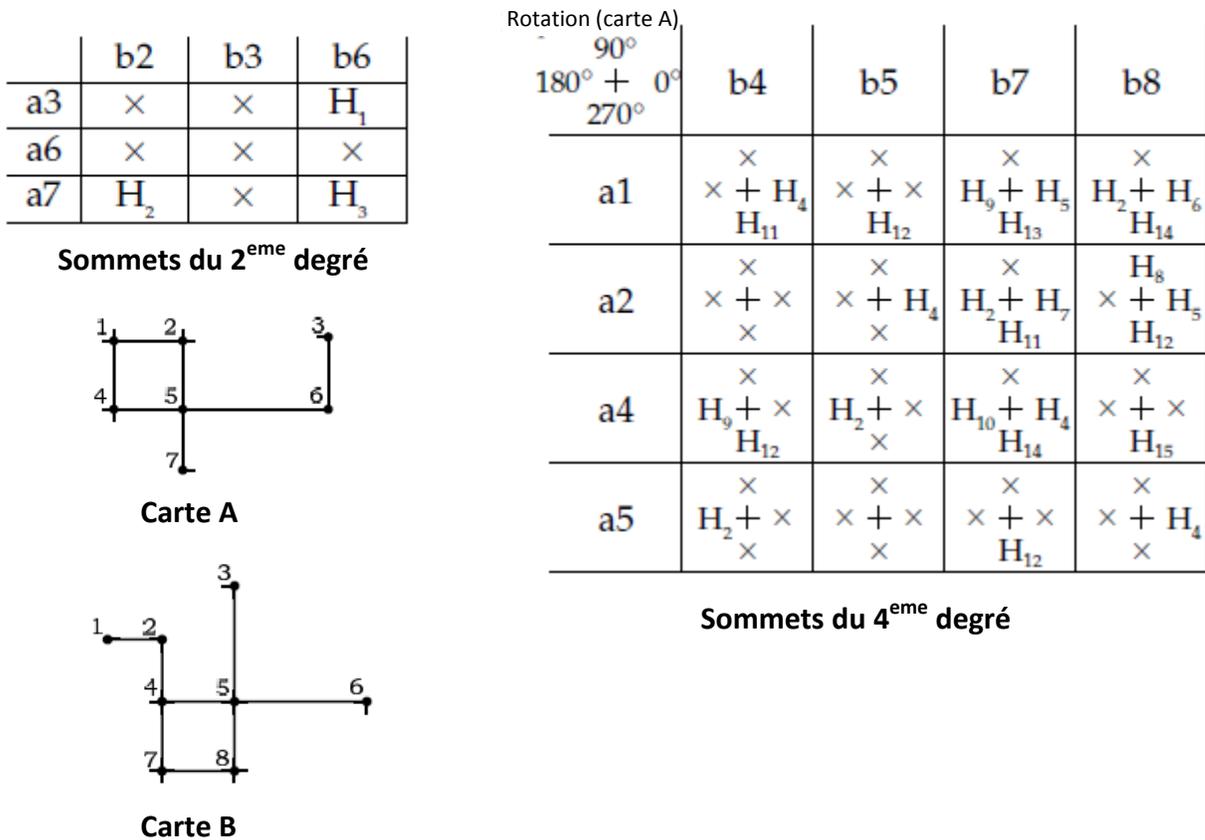
en entier.

Les sommets peuvent être examinés en utilisant les mêmes critères et fonctions de similarité employées pour former les paires initiales. Les arêtes peuvent également avoir des attributs exacts et inexacts connus. Cependant, ils auront simplement la longueur du trajet qui doit être comparée en utilisant une fonction de similarité. Nous mettons un certain seuil pour décider si deux arêtes sont similaires, si c'est le cas, on les ajoute à l'appariement.

Nos hypothèses initiales sont toutes les appariements uniques qui survivent au processus de l'augmentation. Cependant l'approche décrite ci-dessus crée des appariements doubles, il n'est pas difficile de produire et d'augmenter seulement les appariements uniques ; voir l'exemple ci-dessous. On note qu'un sous-graphe dans une carte peut être apparié aux multiples sous-graphes dans l'autre carte (des sous hypothèses séparées), mais une paire de sommets appariés (avec une correspondance d'arête donnée) peut apparaître dans une seule hypothèse.

### **4.2.3 Un exemple**

Pour un exemple sur la génération d'une hypothèse, nous utilisons deux cartes d'un monde rectiligne, représenté dans Fig 4.1. La prétention rectiligne du monde implique que nous savons l'orientation exacte des chemins partant de chaque sommet (relative aux coordonnées du robot). Pour la simplicité, on suppose également que le monde est statique, ainsi le degré de sommets doit s'apparier exactement. Dans ces cartes, les sommets du degré deux et trois peuvent s'apparier seulement avec une paire d'arête simple, alors que les sommets du degré quatre peuvent s'apparier avec quatre paires différentes possibles. Les arêtes sont comparés en utilisant la fonction  $\Phi$  discutée dans l'annexe B; une valeur de seuil différencie entre les appariements des arêtes acceptables et inacceptables.



**FIG 4.1 : Exemple de génération d'hypothèse pour deux cartes d'un monde rectiligne. Les sommets du 2<sup>ème</sup> degré ont seulement un appariement d'arêtes, tandis que les sommets du 4<sup>ème</sup> degré ont quatre correspondances possibles d'arêtes, énuméré en termes d'orientation relative de 0, 90, 180, et 270 degrés. Un « × » indique qu'il n'y a aucune hypothèse valide avec cet appariement de sommet. Il y a 15 hypothèses qui résultent de ces deux cartes.**

Pour construire toutes les hypothèses uniques, nous créons des tables représentant toutes les combinaisons possibles des paires des sommets simples et les correspondances des arêtes incidentes. Alors on augmente chacun de ces appariements en ajoutant les arêtes incidentes et les sommets compatibles. Nous constaterons parfois qu'un appariement est contradictoire, le plus souvent en raison de non correspondance du degré de sommet. Bien que cet appariement doit être jetée, on l'augmente d'abord

autant que possible de sorte que les paires de sommet associées puissent être marquées comme incompatibles. Ceci évite de ré-augmenter le même appariement avec une correspondance de paire de sommet simple.

Par exemple, dans la Fig 4.1, en essayant d'apparier a6-b3, on augmente l'appariement pour inclure a5-b5. Nous pouvons alors voir que les sommets a7-b6 ne s'apparie pas dans les orientations des arêtes relatives (et aussi que les bords a5-a7 et b5-b6 ne s'appariassent pas en raison des longueurs différentes). Cependant, nous augmentons l'appariement pour inclure a2-b4, a1-b7, et a4-b8, et puis on marque tous ces paires comme inadmissibles.

Cependant, on note que les sommets a7-b6 peuvent encore être appareillés (et en fait, mener à une hypothèse valide).

### **4.3 L'évaluation des transformations et le groupement**

Nos hypothèses se composent maintenant de sous-graphes reliés et que la liaison est un appariement maximal.

Ces appariements représentent une seule région de chevauchement, mais les deux cartes peuvent avoir plusieurs régions de chevauchements séparés. Dans cette phase de l'algorithme, on considère la relation géométrique entre les hypothèses.

#### **4.3.1 Estimation de transformation**

Afin d'estimer une transformation géométrique, on doit d'abord inclure les sommets des cartes dans le plan. La situation est identique à celle discutée dans la section 3.6 - bien que les cartes puissent être structurellement conformes, elles pourraient contenir des mesures qui ne sont pas géométriquement conformes en raison des erreurs. Comme mentionné, ce problème a été abordé par plusieurs chercheurs [15, 21, 36]. L'une des méthodes se suffit ; les coordonnées de référence pour les sommets peuvent être placées arbitrairement.

Maintenant on peut estimer une transformation (translation et rotation) pour chaque hypothèse en utilisant les correspondances des sommets de l'hypothèse. Dans le recalage (technique en traitement d'image), ceci serait fait en utilisant itérativement les moindres carrés.

#### **4.3.2 Groupement**

On groupe des hypothèses dans des groupes pour déterminer quelles sont les hypothèses conformes les unes avec les autres. Le groupement est fait sur la base des transformations estimées de l'hypothèse. L'angle de rotation de chaque hypothèse devrait être mesuré pour permettre la comparaison significative de la distance entre les transformations.

Le groupement exige une certaine distance seuil pour de déterminer si deux transformations sont assez proches pour être compatibles. Cette distance peut être définie par rapport à la carte, par exemple, une fraction de la longueur minimum d'une arête dans l'hypothèse. Il y a une variété de techniques pour le groupement ; nous avons employé une méthode de groupement simple dans notre exécution, comme suit :

1. Commencer par chaque hypothèse dans son propre groupe.
2. Comparer toutes les paires de groupes et marquer la paire qui est la plus proche.
3. Comparer la distance de cette paire à la distance seuil :
  - Si la distance est moins que le seuil, grouper la paire dans un groupe simple et revenir à l'étape 2.
  - Si la distance est plus grande que le seuil, le groupement est complet.

Une fois que nous avons un ensemble de groupe d'hypothèse, on ordonne ces groupes par leurs « qualité ». Ceci n'est pas direct sans l'information sur la taille, la

complexité, et la similarité de l'environnement. Nous proposons l'heuristique et les méthodes suivantes :

- Les appariements avec seulement un ou deux paires de sommet ne sont pas significatifs.
- Le nombre total des sommets est un bon indicateur primaire sur la qualité du groupe.
- La quantité d'erreur est un bon indicateur secondaire de qualité.
- Le nombre et la taille d'hypothèses dans un groupe peuvent être employés comme indicateur secondaire de qualité ; pour les groupes de mêmes tailles, ceux qui se composent de peu d'hypothèses sont plus souhaitables.
- Il y a toujours une certaine différence entre la taille et la qualité d'un groupe.

Dans notre exemple, nous avons ordonné les groupes d'abord par leurs taille et puis par l'erreur carrée totale.

#### **4.3.3 Suite de l'exemple**

Nous pouvons trouver la vraisemblance maximum sous un modèle d'erreur gaussien en traitant chaque arête comme un ressort dont la longueur nominale est la distance mesurée et dont la constante de ressort est la réciproque de la variance de mesure.

Pour inclure des sommets sur l'axe des abscisses, nous traitons chaque arête horizontale comme ressort. Pour l'arête  $i$ , la force de ressort est

$$F_i = -\frac{1}{\sigma_i^2}(x_j - x_k - l_k)$$

Ici,  $x_j$  et  $x_k$  sont les x-coordonnés des points finaux de l'arête ( $x_j > x_k$ ), et  $l_k$  est la longueur mesurée de l'arête avec  $\sigma_i^2$  sa variance. Une force  $F_i$  agit à  $x_j$ , et une force  $-F_i$

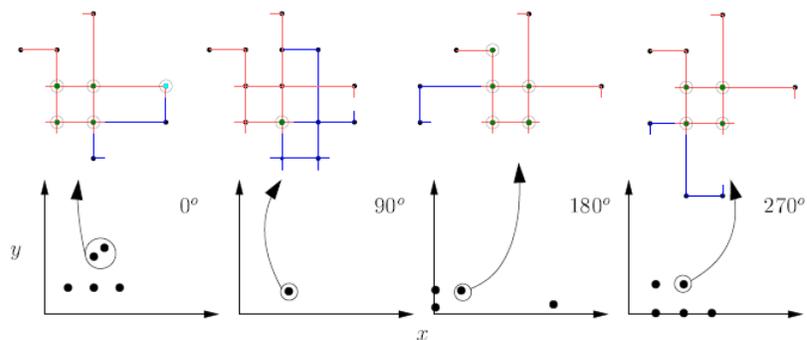
agit à  $x_k$ . Nous écrivons alors une équation pour l'équilibre de force pour chaque coordonné  $x$ ; une coordonné  $x$  peut être arbitrairement choisi d'être  $x=0$ , et aucune équation d'équilibre n'est nécessaire là. S'il y a  $n$  coordonnés  $x$ , on a alors un système linéaires de  $n-1$  équations avec  $n-1$  inconnus ce qui est facilement résolu par les méthodes standard. Le processus est répété avec les coordonnés  $y$  des points finaux des arêtes.

Nous estimons une transformation pour chaque hypothèse en utilisant une version bidimensionnelle de l'algorithme de recalage rigide écrit par Fitzpatrick et al. [20]. Ceci implique de calculer la valeur de décomposition singulière de la matrice de covariance de taille deux-par-deux.

Le schéma 4.2 montre les groupes résultats de notre exemple. En raison de la condition « rectiligne », l'espace de transformation se compose d'un espace de transformations bidimensionnel pour chacune des quatre rotations possibles.

#### 4.4 Quelques mots sur l'implémentation

Si un groupe d'hypothèse est considéré comme correct, il est assez simple pour fusionner les deux cartes. Les évaluations des longueurs des trajets peuvent être mises à jour en combinant les mesures des deux cartes pour les arêtes correspondantes. Les orientations des arêtes aux sommets correspondants peuvent être fusionnées d'une manière similaire.



**FIG 4.2 : Les espaces de transformation et les groupes exemple pour notre carte**

**d'exemple. Les  $180^\circ$ , le groupe de 5 sommets (d'une hypothèse simple) est le meilleur appariement ; le  $0^\circ$ , le groupe de 5 sommets se composant de deux hypothèses est également un bon appariement. La carte A est montré en bleu (foncé), la carte B est montré en rouge, et les sommets appariés sont entourés.**

Les parties d'une carte non actuellement présentent dans l'autre devraient être ajoutées.

#### **4.4.1 Calculs**

Quand l'algorithme de fusion est utilisé en ligne, les robots peuvent avoir des informations supplémentaires qui peuvent être employées pour fusionner les cartes plus rapidement. Par exemple, si deux robots échangent les cartes seulement quand ils sont dans la portée de communication, nous pouvons lier la recherche des hypothèses aux correspondances entre les sommets près des deux robots.

Après la fusion des cartes pour la première fois, les deux robots peuvent fusionner leurs cartes encore une fois à un autre moment postérieur. Pendant les fusions consécutives, une mise à jour « incrémentale » de la carte peut minimiser le temps de calcul; c'est possible avec un effort minimal. La carte du deuxième robot qui est stockée séparément, peut être mise à jour avec les nouveaux sommets et les nouvelles arêtes ou les sommets modifiés et les arêtes modifiées. On peut employer ces mises à jour pour vérifier et augmenter des hypothèses de la fusion précédente, et pour former de nouvelles hypothèses à partir des nouveaux sommets. Les sommets ou les arêtes modifiées peuvent provoquer de nouvelles hypothèses ou peuvent éliminer des hypothèses de l'appariement précédent. Toutes les hypothèses qui ont survie subissent le reste du processus de fusion.

## CHAPITRE 5

# SIMULATIONS ET RÉSULTATS

Nous avons mis en application l'algorithme de cartographie topologique du chapitre 3 et l'algorithme de fusion des cartes topologique du chapitre 4. En ce chapitre, les expériences que nous avons exécutées et les résultats que nous avons obtenus sont détaillés et discutés.

### 5.1 La cartographie topologique d'un robot simple

L'algorithme de cartographie topologique a été mis en application dans la simulation. Les réalisations simulées satisfont les prétentions exigées par le noyau de l'algorithme de cartographie (voir la section 3.1). Parmi ces prétentions est que le modèle d'erreur d'odométrie du robot est approximativement connu. Les expériences qui ont été exécutées ont satisfait l'acceptation d'un environnement fermé, statique, rectiligne.

La majorité des expériences qui ont été exécutées ont examinées l'algorithme de cartographie de base, et en particulier, la méthode « résonnement par évidence » pour le problème de la fermeture de boucle présentée dans la section 3.3 et les améliorations de la carte discutées dans la section 3.6.

Le modèle d'erreur qui a été employé, suppose que le robot commit une erreur d'odométrie qui suit une distribution gaussienne. On assume que l'écart type de cette distribution, désigné sous le nom de  $\sigma$  dans tout ce chapitre, est proportionnel à la longueur des mesures. Typiquement,  $\sigma$  est calculé en tant que certain pourcentage de la longueur de mesure. Par exemple, sous modèle d'erreur supposant que  $\sigma$  est égal à 7%

de la longueur de mesure, la distribution des erreurs pour un mur mesuré pour être 10m de longueur aura (le mur) un écart type  $\sigma = 0.7 M$ .

Dans tous les essais, la limite de fiabilité spécifique à l'algorithme (le paramètre  $\alpha$ ) était 95%. La valeur de seuil de croyance exigée pour accepter ou rejeter une hypothèse de fermeture de boucle, était 99%.

### **5.1.1 Résultats de la simulation**

Le simulateur de robot est un simulateur simple spécifiquement conçu pour examiner l'algorithme de cartographie avec les mondes rectilignes. Le robot est considéré comme une masse ponctuelle. Les comportements déplacer-vers-mur et suivre-mur ont été implémentés et ils répondent aux exigences discutées dans la section 3.2. Pour le comportement suivre-mur, on suppose que le robot n'éprouve aucune erreur de capteur et peut se déplacer dans une ligne droite tout en suivant les murs. De plus, on suppose que le robot n'éprouve aucune erreur d'orientation.

Le comportement suivre-mur renvoie la longueur et les états de terminaison de chaque segment de mur. L'erreur d'odométrie a été simulée en perturbant les distances calculées avec l'erreur gaussienne avec l'écart type  $\sigma$ . Dans les expériences simulées, l'écart type  $\sigma$  a été varié entre 6% et 14% de la longueur de mesure pour les petites expériences, et entre 4% et 20% pour les expériences plus compliquées. On a utilisé la même valeur  $\sigma$  dans l'algorithme de cartographie pour générer des données bruyantes. Bien que la fonction de croyance utilisée pour la fermeture de boucle soit indépendante de  $\sigma$ , ce n'est pas le cas pour la génération des hypothèses, en raison des valeurs fixes de  $r_0$ , du  $r_{\min}$ , du  $r_{\max}$  et du  $r^*$ .

#### **5.1.1.1 Expériences**

L'algorithme de cartographie de base a été testé dans le simulateur avec un certain nombre de mondes simples. La Fig 5.2 montre quatre de tels mondes par ordre de complexité croissante. Pour ces environnements, le robot a construit un certain

nombre de cartes dans des conditions de plus en plus difficiles (à chaque fois on augmente l'erreur). Les résultats étaient généralement comme prévus : 97% des cartes étaient structurellement corrects après que le robot ait accepté une hypothèse de fermeture-boucle. Si le robot rejette l'hypothèse correcte, la carte est immédiatement marquée incorrecte.

Plusieurs expériences ont été exécutées pour examiner la susceptibilité de la méthode de fermeture de boucle aux hypothèses incorrectes en mondes auto-similaires. Comme discuté dans la section 3.3, ces mondes sont les plus difficiles pour fermer des boucles. L'algorithme de cartographie de base va bien dans la plupart des cas, bien que nous ayons pu construire quelques environnements auto-similaires anormaux pour lesquels l'algorithme prend parfois des décisions incorrectes dans certaines conditions d'erreur au sujet d'une hypothèse de fermeture de boucle.

La Fig 5.3 donne des résultats pour la cartographie d'un monde auto-similaire « spiral ». Un certain nombre de cartes d'essai ont été produites de ce monde sous des divers niveaux d'erreur. L'approche de raisonnement par évidence était réussie en choisissant l'hypothèse correcte de fermeture de boucle dans la plupart des cas. L'hypothèse correcte a été choisie environ 87% du temps dans nos épreuves, dans des conditions extrêmement dures d'erreur, avec un écart type plus que 10% de la longueur de mesure. Généralement l'algorithme va bien avec ces mondes « en spirales » - tant que les spirales ne sont pas extrêmement « étroites ».

La Fig 5.4 donne les résultats de cartographie d'un monde avec « répétition de séquences similaires » des murs.

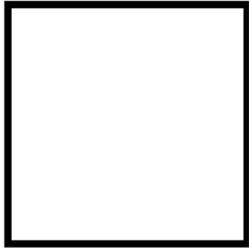
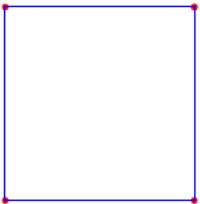
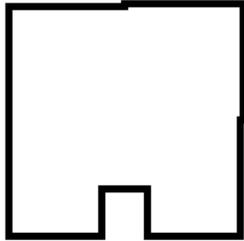
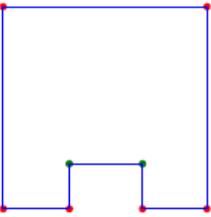
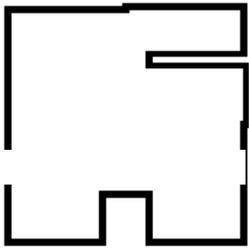
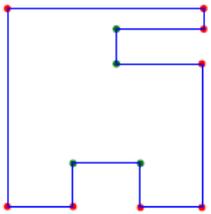
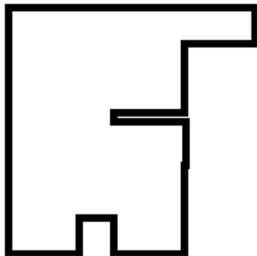
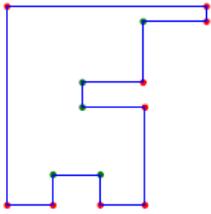
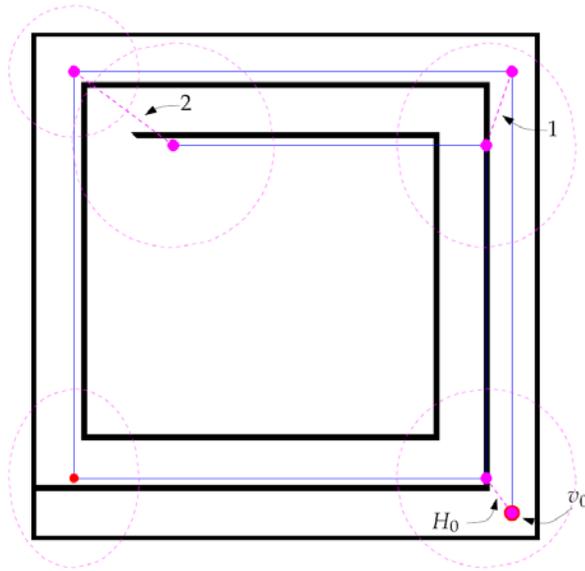
Environnement	Carte	Statistiques		
		$\sigma$	Correcte	Incorrecte
		$\sigma$	Correcte	Incorrecte
		6%	10	0
		8%	10	0
		10%	10	0
		12%	10	0
		14%	10	0
		$\sigma$	Correcte	Incorrecte
		6%	10	0
		8%	10	0
		10%	10	0
		12%	10	0
		14%	9	1
		$\sigma$	Correcte	Incorrecte
		6%	10	0
		8%	10	0
		10%	10	0
		12%	10	0
		14%	8	2
		$\sigma$	Correcte	Incorrecte
		6%	10	0
		8%	10	0
		10%	9	1
		12%	9	1
		14%	9	1

FIG 5.2 : La cartographie des mondes simples simulés. Chaque environnement a été testé 10 fois dans de diverses conditions d'erreur. Les tables montrent la fréquence des cartes correctes (pour ce que la bonne hypothèse de fermeture de boucle a été acceptée) contre la fréquence des cartes incorrectes (pour ce que l'hypothèse fautive a été acceptée ou l'hypothèse correcte a été rejeté)



$\sigma$	Correcte	Incorrecte
4%	10	0
6%	10	0
8%	10	0
10%	9	1
12%	9	1
14%	9	1
16%	8	2
18%	9	1
20%	8	2

Etape	$m(C)$	$m(I)$	$m(\{C,I\})$
0	0.202	0.000	0.798
1	0.007	0.975	0.019
2	0.000	0.999	0.000

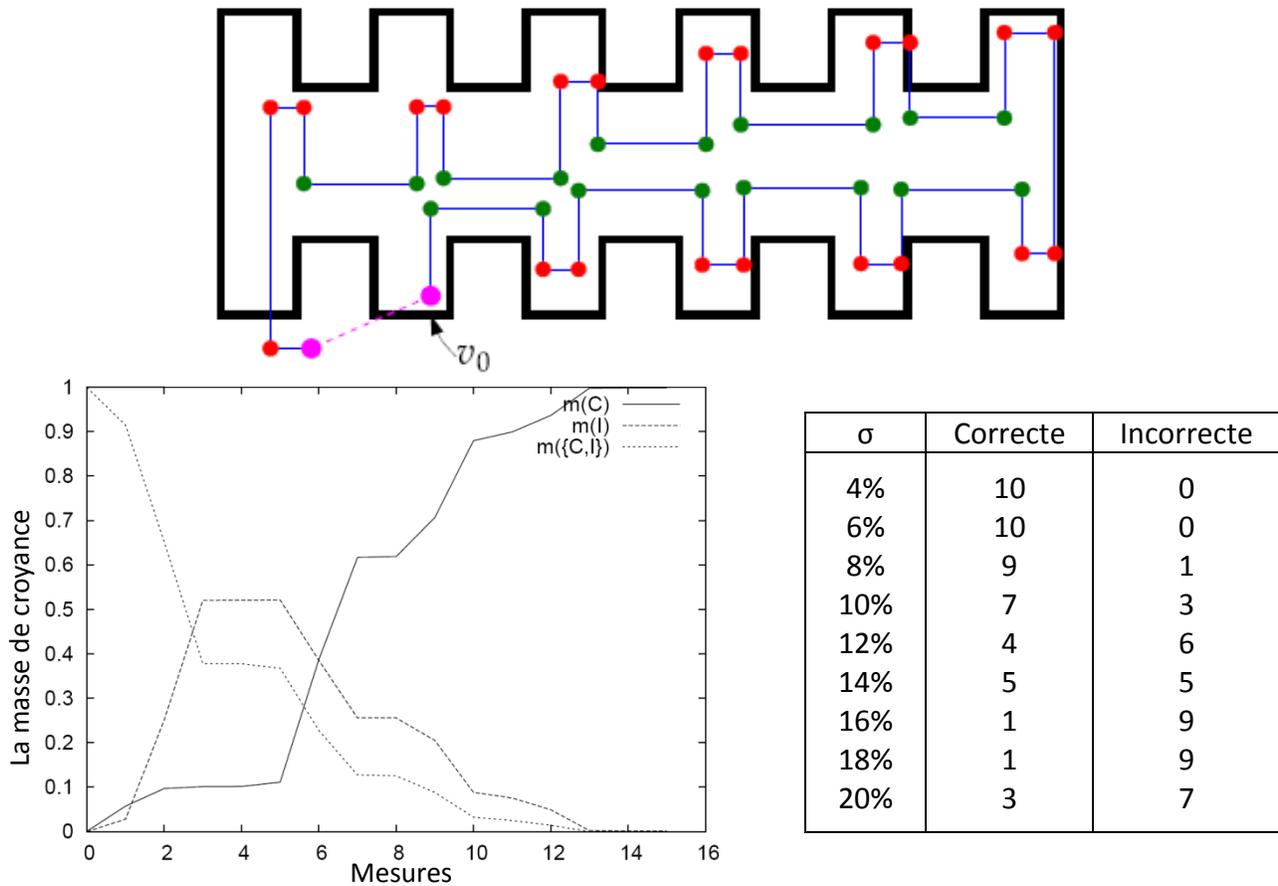
**FIG 5.3 : Construction d'une carte pour un monde « spirale » auto-similaire. Dix tests ont été effectués pour chaque valeur d'erreur (table droite). Dans toutes les épreuves, le robot a commencé à la même configuration initiale, de sorte que le sommet de début a été toujours identique. Un cas représentatif est décrit. Ici, des correspondances de sommet selon une hypothèse incorrecte sont montrées comme lignes tirées entre les sommets. Les limites de confiance de 95% sont montrées comme ellipses sur les endroits de sommet. Le véritable sommet de début est à la droite inférieure extrême (grand point) ; une hypothèse incorrecte de fermeture de boucle a été produite après la fabrication d'un « recouvrement » autour de la spirale. Après la prise de deux mesures supplémentaires de longueur de mur, l'algorithme a tenu 0.99 croyance que l'hypothèse était incorrecte. La table au-dessous de la carte montre les valeurs globales de fonction de croyance après chaque nouvelle mesure. Étape 0 indique les valeurs initiales de fonction de croyance (voir l'annexe B pour plus de détails).**

Ces mondes sont plus difficiles pour l'algorithme de cartographie, et cet environnement en particulier défait l'algorithme souvent. Les difficultés surgissent quand le robot confond une « bonne » hypothèse avec des autres quand il essaye de fermer la boucle.

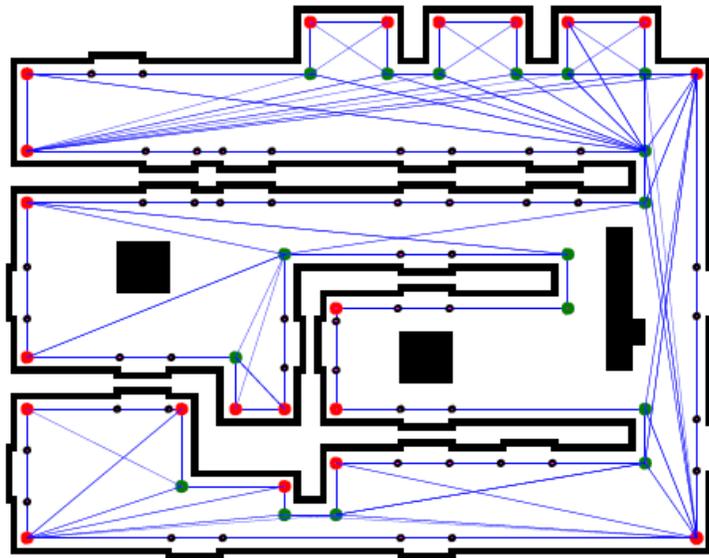
Dans des conditions relativement bénignes, le robot peut généralement fermer la boucle correctement, parce que la distance entre les sommets est plus grande que son accumulation.

En simulant des mondes plus réalistes, comme cela représenté sur la Fig 5.5, l'approche du raisonnement par évidence tend à se comporter mieux. Au-dessus d'un grand nombre de test dans cet environnement, une seule décision incorrecte de fermeture de boucle a été prise. Ceci s'est produit dans une situation où le robot a éprouvé l'erreur de mesure avec un écart type égal à 20% de la véritable longueur de mur. Dans l'autre essai, des ordres de mesures improbables se sont produits de temps en temps ; dans de telles circonstances, des hypothèses correctes sont parfois rejetées, ou des hypothèses incorrectes sont acceptées.

Les « améliorations » à la carte de base discutée dans la section 3.6 ajoutent des chemins entre les sommets de carte « en faisant une incursion » de l'espace vide. Cette approche a été examinée dans la simulation. La Fig 5.6 décrit une carte du monde de la Fig 5.5, avec des améliorations à la carte de base incluse. La figure a été produite avec un robot simulé n'éprouvant aucune erreur de mesure. Des améliorations ont été examinées dans de diverses conditions d'erreur en variant le  $\sigma$ . Comme prévu, pour une erreur angulaire ou odométrique significative, seulement des améliorations entre les sommets à côté des murs longs peuvent être faites. Cependant, juste quelques améliorations sont assez bénéfiques pour améliorer la navigation dans la carte.



**FIG 5.4 : Un environnement qui défait la méthode du raisonnement par évidence pour la fermeture de boucle dans certaines conditions d'erreur. Dix épreuves ont été effectuées pour chaque niveau d'erreur (table). Dans toutes les épreuves, le robot a commencé de la même configuration initiale, de sorte que le sommet de début ait été toujours en tant que décrit (le pire scénario). Un cas représentatif est décrit. Ici, le sommet de début est dans le « bon » qui est en second lieu de la gauche sur le fond. Avant que le robot explore le monde pour atteindre l'extrême gauche au fond, il confond ceci avec le début et produit une hypothèse incorrecte (montrée comme ligne tirée entre les sommets assortis). En raison de l'autosimilarité extrême de la partie de l'environnement contenant le sommet de début, la croyance  $m(C)$  dans l'exactitude de l'hypothèse dépasse le seuil requis  $\beta = 0.99$  avant que le robot rencontre une disparité structurale.**



**FIG 5.6 : Une carte du monde avec des améliorations incluses. Les « îles » n'ont pas été produites. Cette carte a été produite sans erreur afin d'illustrer une carte « idéale ».**

### 5.1.2 Discussion

D'une façon générale, nos expériences simulées indiquent que l'algorithme de cartographie fonctionne comme prévu dans la plupart des cas – il est bien dans 95% des cas pour les mondes non-auto-similaires. La plupart des expériences se sont concentrées sur l'examen de l'approche « résonnement par évidence » pour la fermeture de boucle. Nous avons pu défaire cette approche en construisant des environnements auto-similaires, mais cela a fonctionné bien dans tous les essais qui ont été sur des environnements proches du réel.

Dans les réalisations simulées, le robot éprouve une erreur d'odométrie avec un écart type proportionnel à la distance voyageée. Dans les tests dans les environnements compliqués, nous avons vu que l'approche de fermeture de boucle est sensible à de grandes inexactitudes dans le modèle d'erreur - en particulier, le robot manque des hypothèses valides de fermeture de boucle en raison des limites de confiance sur son

endroit trop-serrées.

L'idée des « améliorations » qui relie les endroits dans la carte de base en utilisant des incursions dans l'espace ouvert a été examinée. L'idée semble saine. Il y a typiquement un grand nombre d'améliorations potentielles, puisque chaque sommet peut habituellement se relier à plusieurs autres. Une amélioration possible pourrait être d'essayer d'explorer des améliorations multiples simultanément : souvent, plusieurs améliorations traversent l'espace qui peut « être balayé » par les capteurs du robot pendant qu'il explore une amélioration similaire.

Nous n'avons pas intensivement examiné le comportement suivre-couloir ou l'idée des « portails » entre les couloirs serrés et les espaces ouverts. Bien que ces idées ne soient pas centrales à l'algorithme de cartographie, elles peuvent améliorer l'efficacité du mapping d'une manière significative.

## **5.2 Fusion des cartes topologiques**

L'algorithme de fusion des cartes topologiques a été mis en application pour les mondes rectilignes, et examinée dans des environnements fabriqués à la main et aléatoirement produits, avec des cartes construites sous des divers niveaux d'erreur. Les résultats indiquent que l'algorithme de fusion est très efficace, même pour de grandes cartes et un petit chevauchement.

Nos expériences supposent que le robot peut déterminer le degré de sommets de carte lors de les découvrir. Ainsi, dans toutes les expériences de fusion des cartes, deux sommets sont considérés comme des appariements initiaux potentiels s'ils sont du même type et degré. Ainsi, parce que pour les sommets du degré quatre, chaque appariement rapporte quatre correspondances possibles - une pour chaque orientation possible dans un monde rectiligne. Les sommets du degré moins de quatre rapportent seulement une seule correspondance possible.

### 5.2.1 Résultats de simulation

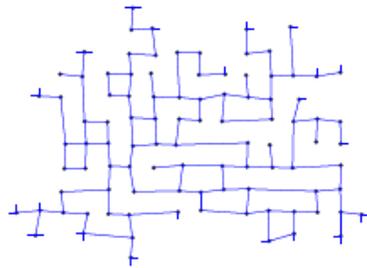
Le simulateur, mis en application séparément du simulateur précédent, produit des mondes grands et aléatoires avec des boucles comme les labyrinthes. Ensuite il produit des cartes partielles à partir des mondes en sélectionnant les positions initiales aléatoirement.

L'erreur est présentée dans les cartes en troublant aléatoirement les endroits de sommet selon une distribution gaussienne avec un écart type configurable. Sur option, l'erreur présentée en perturbant un sommet « est propagée » par les cartes en perturbant pareillement ses sommets voisins, afin de simuler des conditions d'erreur réelles.

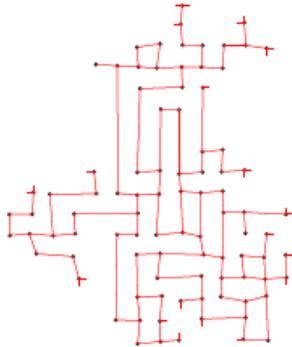
La propagation d'erreur est « extérieure » de la racine de l'arbre de recherche en largeur duquel la carte a été produite (une commande absolue des sommets est spécifiée pour la perturbation de sorte que la propagation d'erreur n'affecte pas des sommets) « déjà-visités ».

Après avoir produit des cartes partielles, l'algorithme de fusion parcourt ces cartes pour produire des correspondances.

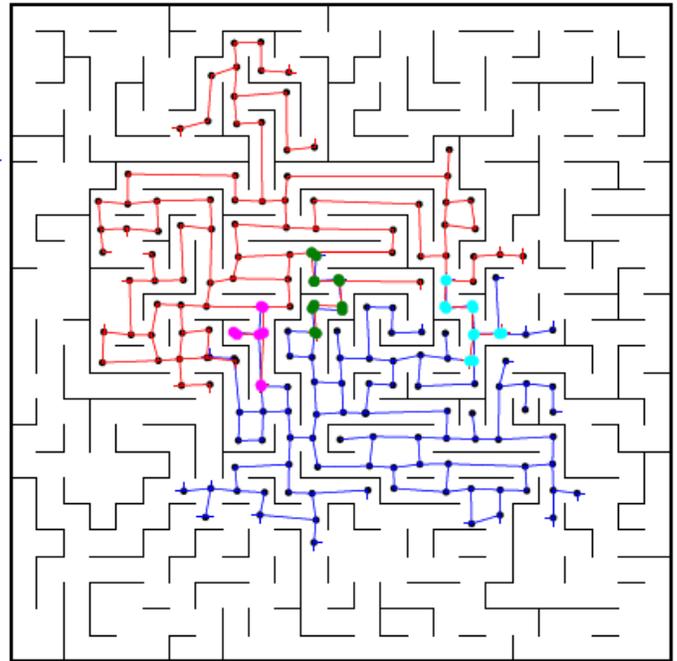
La Fig 5.9 montre l'appariement trouvé pour deux cartes aléatoirement produites. Les deux cartes sont montrées après appariement. Les cartes se chevauchent dans plusieurs endroits « structurellement débranchés » ; le résultat final du processus est un groupe de trois hypothèses.



Carte A



Carte B



Correspondance de sommet	3918
Hypothèse après augmentation	72
Groupes	69
Meilleur groupe	3 hypothèses, 16 sommets
Temps d'exécution	0.04 secondes

**FIG 5.9 : Résultats de fusion des cartes simulées aléatoirement produites. Des sommets appariés sont montrés avec de grands points, et les différents appariements ont différentes couleurs/nuances. Le meilleur appariement se produit quand la carte B est approximativement tournée  $-90^{\circ}$ , et se compose de trois hypothèses séparées qui sont groupées ensemble.**

Un plus grand exemple, produit en utilisant plus de bruit et avec une certaine propagation d'erreur dans les cartes, est montré sur la Fig 5.10. Ici, les deux cartes, ont plus de 450 sommets. Les deux cartes sont correctement appariées avec un groupe de

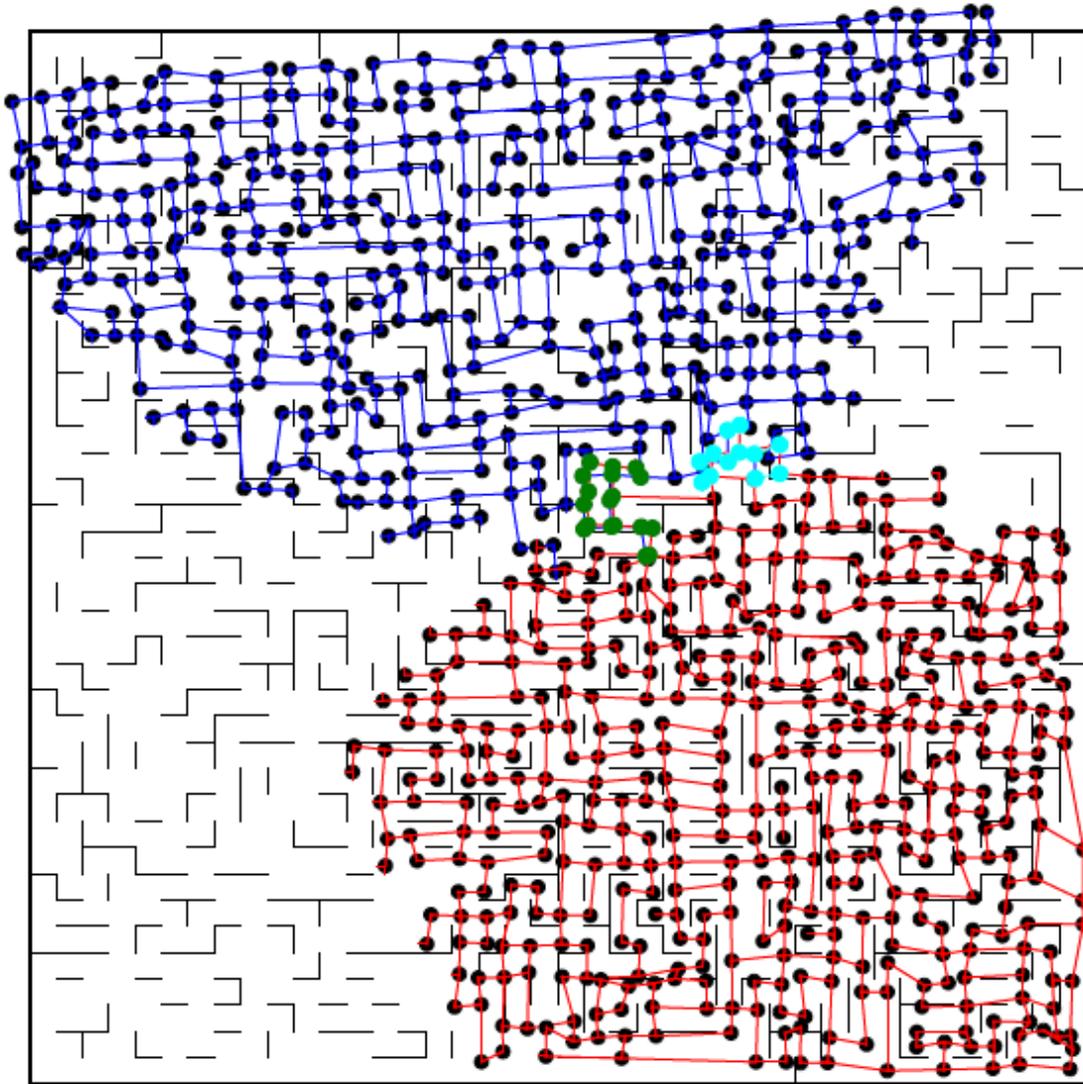
deux hypothèses. Le temps total d'exécution nécessaire pour produire l'appariement était sous la moitié de la seconde. En raison des erreurs de mesure, une petite inclinaison s'est produite après la transformation.

### **5.2.3 Discussion**

L'algorithme de fusion des cartes a prouvé son succès pour les cartes simulées que nous avons pu examiner. Beaucoup de ceci est due à la correspondance structurale effectuée par l'algorithme. Bien que structurellement les mondes auto-similaires (comme ceux qui posent des difficultés pour l'algorithme de cartographie) puissent parfois mener aux « disparités ».

Pour de grandes cartes avec plus que 100 sommets, il y avait généralement plusieurs milliers de correspondances de simple-sommets. Après le processus de croissance topologique, le nombre d'hypothèses restantes était moins de 100 ; typiquement, il y avait seulement une ou deux grandes hypothèses contenant plus de trois sommets.

Les transformations géométriques prévues pour les groupes corrects présentent parfois une « inclinaison » significative dans la carte fusionnée, comme vu sur la Fig 5.10. L'inclinaison est plus grande pour les cartes avec des petits chevauchements ; les cartes avec un grand chevauchement fournissent plus de données pour l'usage en estimation de la transformation. On note que les transformations estimées peuvent être améliorées si les différents robots augmentent leurs propres cartes et comparent leurs nouvelles données à la carte fusionnée.



Correspondance de sommet	98372
Hypothèse après augmentation	80
Groupes	77
Meilleur groupe	2 hypothèses, 15 sommets
Temps d'exécution	0.43 secondes

**FIG 5.10 : Un appariement entre de grandes cartes, chacun avec approximativement 450 sommets. L'appariement correct est un groupe de deux hypothèses.**

## **CHAPTER 6**

# **CONCLUSIONS ET PERSPECTIVES**

Ce mémoire a examiné deux aspects de la cartographie multi-robot et l'exploration : cartographie topologique d'un robot simple et la fusion des cartes topologiques. Il s'est concentré principalement sur l'utilisation des robots limités - robots capables seulement de la détection à courte portée. Ces robots sont peu coûteux et donc potentiellement jetables, ce qui rends ces robots particulièrement utiles dans des situations qui nécessitent le mapping par de grandes équipes de petits robots mobiles.

### **6.1 La cartographie topologique par un robot simple**

Dans ce mémoire, on a développé un algorithme de cartographie topologique pour les robots limités qui permet à des différents robots de construire des cartes des environnements d'intérieur. L'algorithme adopte une approche de raisonnement par évidence, basée sur la théorie de Dempster-Shafer de preuve, pour le problème de

« fermeture de boucle » - pour identifier si le robot est revenu à un endroit qu'il a déjà visité ou non. En plus, l'algorithme présente des idées pour incorporer les espaces ouverts dans les cartes topologiques, en appliquant des améliorations à notre carte. Ces améliorations sont basées sur les « incursions » dans les espaces ouverts.

Les expériences, ont vérifiées l'efficacité de l'approche. L'idée des « améliorations » basées sur les incursions en espace ouvert a été vérifiée en simulation. Cependant en mondes auto-similaires la méthode de fermeture de boucle prend parfois des décisions incorrectes. Dans nos essais, l'hypothèse correcte de fermeture de boucle a été acceptée plus de 95% de temps dans les environnements simulés.

## **6.2 Fusion des cartes topologiques**

Ce mémoire a également présenté un algorithme pour fusionner les cartes topologiques de deux robots sans nécessité d'une référence commune. L'algorithme emploie la structure des cartes topologiques pour trouver un ensemble d'appariement probables, et puis emploie les transformations géométriques de ces hypothèses pour les grouper dans des groupes denses. Les heuristiques possibles pour déterminer la qualité des groupes ont été décrites.

L'algorithme de fusion des cartes a été démontré sur les cartes simulées. Dans nos expériences, des cartes avec un chevauchement significatif ont été toujours fusionnées correctement. Même des cartes avec un chevauchement minimal sont souvent fusionnées correctement. La performance de l'algorithme est tout à fait bonne ; pour deux cartes avec environ 450 sommets chacune, les correspondances correctes entre les sommets ont été trouvées en moins d'une demi-seconde.

## **6.3 Perspectives**

Les deux algorithmes présentés ici sont supposé qu'ils traitent seulement les environnements statiques. L'élargissement des deux pour manipuler les

environnements dynamiques améliorera considérablement leurs performances aux situations réelles. Le problème des environnements dynamiques est liée au problème des erreurs et des contradictions structurales dans les cartes ; la détection et la résolution de ces erreurs est particulièrement importante, puisqu'elles sont trop probables de se produire quand des robots limités sont utilisés.

Il y a d'autres questions fondamentales qui devraient être explorées :

- Comment la qualité et la performance de la cartographie sont affectées : par les limites des capteurs? Par l'erreur de capteur ? Par la complexité du monde ?
- Quelles sont les avantages fournis par des prétentions spécifiques sur le monde?
- Que peut être fait pour construire des cartes dans un monde incurvé?

L'algorithme de fusion des cartes topologiques fonctionne bien, mais il pourrait être amélioré. En particulier, il serait utile de rendre l'algorithme distribué. Ainsi, l'algorithme pourrait être exécuté en ligne même par des robots limités en puissances de calcul.

# BIBLIOGRAPHIE

- [1] M.A. Bender, A. Fernandez, D. Ron, A. Sahai, and S. Vadhan. "The power of a pebble: Exploring and mapping directed graphs". In Proceedings of the Annual Symposium on Foundations of Computer Science, pp. 269–278, 1998.
- [2] P.J. Besl and N.D. McKay. "A method for registration of 3-D shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence", 14(2):pp.239–256, February 1992.
- [3] L.G. Brown. "A survey of image registration techniques". ACM Computing Surveys, 24(4):pp.325–376, 1992.
- [4] H. Bunke. "On a relation between graph edit distance and maximum common subgraph". Pattern Recognition Letters, 18(8):pp.689–694, 1997.
- [5] H. Bunke and G. Allermann. "Inexact graph matching for structural pattern recognition". Pattern Recognition Letters, 1(4):pp.245–253, 1983.
- [6] Z.J. Butler, A.A. Rizzi, and R.L. Hollis. "Distributed coverage of rectilinear environments". In Proceedings of the Workshop on the Algorithmic Foundations of Robotics, pp. 51–62. A.K. Peters, January 2001.
- [7] R. Chatila and J.-P. Laumond. "Position referencing and consistent world modeling for mobile robots". In Proceedings of the 1985 IEEE International Conference on Robotics & Automation, pp. 138–145, St. Louis, March 1985.
- [8] H. Choset and K. Nagatani. "Topological simultaneous localization and mapping (SLAM): Toward exact localization without explicit

localization". IEEE Transactions on Robotics & Automation, 17(2), April 2001.

- [9] D. Conte, P. Foggia, C. Sansone, and M. Vento. "Thirty years of graph matching in pattern recognition". International Journal of Pattern Recognition and Artificial Intelligence, 2004.
- [10] I. Cox and J. Leonard. "Modeling a dynamic environment using a Bayesian multiple hypothesis approach". Artificial Intelligence, 66:pp.311–344, 1994.
- [11] G. Dedeoglu and G.S. Sukhatme. "Landmark-based matching algorithm for cooperative mapping by autonomous robots". In Proceedings of the 2000, pp. 251–260, Knoxville, Tennessee, October 2000.
- [12] M. DiMarco, A. Garulli, A. Giannitrapani, and A. Vicino. "A set theoretic approach to dynamic robot localization and mapping". Autonomous Robots, 16(1):pp.23–47, January 2004.
- [13] K.L. Doty and S.L. Seed. "Autonomous agent map construction in unknown enclosed environments". In Proceedings of MLC-COLT Workshop on Robot Learning, pp. 47–55, New Brunswick, N.J., July 1994.
- [14] T. Duckett, S. Marsland, and J. Shapiro. "Learning globally consistent maps by relaxation". In Proceedings of the 2000 IEEE International Conference on Robotics & Automation, volume 4, pp 3841–3846, 2000.
- [15] T. Duckett and A. Saffiotti. "Building globally consistent gridmaps from topologies". In Proceedings of the 6th International IFAC Symposium on Robot Control (SYROCO), pp. 357–361, Wien, Austria, 2000.
- [16] G. Dudek, M. Jenkin, E. Milos, and D. Wilkes. "Topological exploration with multiple robots". In Proceedings of the Seventh International Symposium on Robotics with Applications (ISORA '98), Alaska, May 1998.

- [17] P.J. Durand, R. Pasari, J.W. Baker, and C. Tsai. "An efficient algorithm for similarity analysis of molecules". *Internet Journal of Chemistry*, 2, 1999.
- [18] H. Feder, J. Leonard, and C. Smith. "Adaptive mobile robot navigation and mapping". *International Journal of Robotics Research*, 18(7):pp.650–668, July 1999.
- [19] J.W. Fenwick, P.M. Newman, and J.J. Leonard. "Cooperative concurrent mapping and localization". In *Proceedings of the 2002 IEEE International Conference on Robotics & Automation*, May 2002.
- [20] J.M. Fitzpatrick, D.L.G. Hill, and C.R. Maurer, Jr. Image registration. In Milan Sonka and J.M. Fitzpatrick, editors, *Handbook of Medical Imaging*, volume 2: Medical Image Processing and Analysis, chapter 8. SPIE, 2000.
- [21] M. Golfarelli, D. Maio, and S. Rizzi. "Elastic correction of dead-reckoning errors in map building". In *Proceedings of the 1998 IEEE/RSJ International Conference on intelligent Robots & Systems*, pp. 905–911, B.C., Canada, 1998.
- [22] R. Grabowski, L. Navarro-Serment, C. Paredis, and P. Khosla. "Heterogeneous teams of modular robots for mapping and exploration". *Autonomous Robots*, 8(3):pp.293–308, 1999.
- [23] A. Howard. "Multi-robot mapping using manifold representations". In *Proceedings of the 2004 IEEE International Conference on Robotics & Automation*, April 2004.
- [24] W.H. Huang and K.R. Beevers. "Topological map merging". 7th *International Symposium on Distributed Autonomous Robotic Systems (DARS 2004)*, 2004.
- [25] W.H. Huang and K.R. Beevers. "Topological mapping with sensing-limited robots". 6th *International Workshop on the Algorithmic Foundations of Robotics (WAFR 2004)*, 2004.

- [26] J. Jennings, C. Kirkwood-Watts, and C. Tanis. "Distributed map-making and navigation in dynamic environments". In Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots & Systems, volume 3, pp. 1695–1701, Victoria, B.C., Canada, October 1998.
- [27] J. Ko, B. Stewart, D. Fox, K. Konolige, and B. Limketkai. "A practical, decision-theoretic approach to multi-robot mapping and exploration". In Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots & Systems, 2003.
- [28] K. Konolige. "Improved occupancy grids for map building. *Autonomous Robots*, 4(4), December 1997.
- [29] K. Konolige, D. Fox, B. Limketkai, J. Ko, and B. Stewart. "Map merging for distributed robot navigation". In Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots & Systems, volume 1, pp. 212–217, October 2003.
- [30] B. Kuipers. "Modeling spatial knowledge". *Cognitive Science*, 2:pp.129–153, 1978.
- [31] B. Kuipers. The spatial semantic hierarchy. *Artificial Intelligence*, 119(1-2):pp.191–233, 2000.
- [32] B. Kuipers and P. Beeson. "Bootstrap learning for place recognition". In Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-02), Edmonton, Canada, 2002.
- [33] B.J. Kuipers and Y.-T. Byun. "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations". *Journal of Robotics and Autonomous Systems*, 8:pp.47–63, 1991.
- [34] J.J. Leonard and H.J.S. Feder. "A computationally efficient method for large-scale concurrent mapping and localization". In J. Hollerbach and D. Koditschek, editors, *International Symposium on Robotics Research*, pp. 169–176, 1999.

- [35] B. Limketkai, R. Biswas, and S. Thrun. "Learning occupancy grid maps of non-stationary objects with mobile robots in indoor environments". In Proceedings of the 2002 International Symposium on Experimental Robotics (ISER 2002), Italy, July 2002.
- [36] F. Lu and E. Milius. "Globally consistent range scan alignment for environment mapping". *Autonomous Robots*, 4(4):pp.333–349, October 1997.
- [37] B. Luo and E.R. Hancock. "Structural graph matching using the EM algorithm and singular value decomposition". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10), October 2001.
- [38] J.J. McGregor. Backtrack search algorithms and the maximal common subgraph problem. *Software Practice and Experience*, 12(1):pp.23–34, 1982.
- [39] B.T. Messmer and H. Bunke. "A new algorithm for error-tolerant subgraph isomorphism detection". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5), May 1998.
- [40] H.P. Moravec and A. Elfes. "High resolution maps from wide angle sonar". In Proceedings of the 1985 IEEE International Conference on Robotics & Automation, pp. 116–121, 1985.
- [41] R.R. Murphy. "Dempster-Shafer theory for sensor fusion in autonomous mobile robots". *IEEE Transactions on Robotics & Automation*, 14(2):pp.197–206, April 1998.
- [42] D. Pagac, E.M. Nebot, and H. Durrant-Whyte. "An evidential approach to map-building for autonomous vehicles". *IEEE Transactions on Robotics & Automation*, 14(4):pp.623–629, August 1998.
- [43] E. Remolina and B. Kuipers. "Towards a general theory of topological maps". *Artificial Intelligence Journal*, 2002.
- [44] N. Roy and S. Thrun. "Coastal navigation with mobile robots". *Advances in Neural Processing Systems*, 12:pp.1043–1049, 1999.

- [45] P.E. Rybski, F. Zacharias, J.-F. Lett, O. Masoud, M. Gini, and N. Papanikolopoulos. "Using visual features to build topological maps of indoor environments". In Proceedings of the 2003 IEEE International Conference on Robotics & Automation, Taipei, Taiwan, September 2003.
- [46] B. Schiele and J.L. Crowley. "A comparison of position estimation techniques using occupancy grids". In Proceedings of the 1994 IEEE International Conference on Robotics & Automation, pp. 1628–1634, San Diego, CA, May 1994.
- [47] K. Sents and S. Ferson. "Combination of evidence in Dempster-Shafer theory". Technical Report SAND2002-0835, Sandia National Laboratories & Binghamton University, April 2002.
- [48] G. Shafer. "Perspectives on the theory and practice of belief functions". *International Journal of Approximate Reasoning*, 4:pp.323–362, 1990.
- [49] Glenn Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.
- [50] H. Shatkay and L.P. Kaelbling. "Learning topological maps with weak local odometric information". In Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, 1997.
- [51] H. Shen, C.V. Stewart, B. Roysam, G. Lin, and H.L. Tanenbaum. "Frame-rate spatial referencing based on invariant indexing and alignment with application to on-line retinal image registration". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(3), March 2003.
- [52] R. Smith, M. Self, and P. Cheeseman. "Estimating uncertain spatial relationships in robotics". In I. Cox and G. Wilfong, editors, *Autonomous Robot Vehicles*. Springer-Verlag, 1990.
- [53] C.V. Stewart, C.-L. Tsai, and A. Perera. "A view-based approach to registration: Theory and application to vascular image registration". In

Proceedings of the International Conference on Information Processing in Medical Imaging (IPMI), May 2003.

- [54] C.V. Stewart, C.-L. Tsai, and B. Roysam. "The dual-bootstrap iterative closest point algorithm with application to retinal image registration". Technical Report RPI-CS-TR 02-9, RPI Department of Computer Science, Troy, NY, June 2002.
- [55] A. Stroupe, M. Martin, and T. Balch. "Distributed sensor fusion for object position estimation by multi-robot systems". In Proceedings of the 2001 IEEE International Conference on Robotics & Automation, 2001.
- [56] J. Sudano. "Pignistic probability transforms for mixes of low- and high-probability events". In Proceedings of the 4th International Conference on Information Fusion, pp. 23–27, Montreal, August 2001.
- [57] S. Thrun. "A probabilistic online mapping algorithm for teams of mobile robots". *International Journal of Robotics Research*, 20(5):pp.335–363, 2001.
- [58] S. Thrun and A. Bücken. "Integrating grid-based and topological maps for mobile robot navigation". In Proceedings of the Thirteenth National Conference on Artificial Intelligence, pp. 944–950, 1996.
- [59] S. Thrun, D. Fox, and W. Burgard. "Probabilistic mapping of an environment by a mobile robot". In Proceedings of the 1998 IEEE International Conference on Robotics & Automation, pages 1546–1551, 1998.
- [60] S. Thrun, J. Gutmann, D. Fox, W. Burgard, and B. Kuipers. "Integrating topological and metric maps for mobile robot navigation: A statistical approach". In Proceedings of the National Conference on Artificial Intelligence (AAAI), 1998.

- [61] Sebastian Thrun. Robotic mapping: A survey. Technical Report CMU-CS-02-111, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, February 2002.
- [62] N. Tomatis, I. Nourbakhsh, and R. Siegwart. "Combining topological and metric: a natural integration for simultaneous localization and map building". In Proceedings of the Fourth European Workshop on Advanced Mobile Robots (Eurobot 2001), pp. 19–21, Lund, Sweden, September 2001.
- [63] N. Tomatis, I. Nourbakhsh, and R. Siegwart. "Hybrid simultaneous localization and map building: closing the loop with multi-hypothesis tracking". In Proceedings of the 2002 IEEE International Conference on Robotics & Automation, May 2002.
- [64] B. Tovar, S. LaValle, and R. Murrieta. "Optimal navigation and object finding without geometric maps or localization". In Proceedings of the 2003 IEEE International Conference on Robotics & Automation, Taipei, Taiwan, September 2003.
- [65] Ray A. Waller. Statistics: An introduction to numerical reasoning. Holden-Day, San Francisco, 1979.
- [66] R.C. Wilson and E.R. Hancock. "Structural matching by discrete relaxation". IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(6), June 1997.

# ANNEXE A

## • LA THÉORIE DE PREUVE DE DEMPSTER-SHAFER

La théorie de Dempster-Shafer de preuve, d'abord présentée par Shafer en 1976 [49], fournit un cadre pour la représentation mathématique de l'incertitude ce qui la diffère de la théorie des probabilités traditionnelle de plusieurs manières principales :

- La théorie de Dempster-Shafer est capable de représenter l'« ignorance » - un manque de croyance dans l'exactitude de n'importe quelle hypothèse.
- Elle permet l'attribution de la masse de croyance aux ensembles des possibilités mutuellement exclusives - pas simplement les possibilités individuelles.
- En raison de sa capacité de représenter l'ignorance, la théorie de Dempster-Shafer n'exige aucune connaissance a priori sur le monde ; en présence d'aucune connaissance, toute la masse de croyance est assignée à l'ignorance.

Dans la théorie de Dempster-Shafer, l'ensemble des possibilités mutuellement exclusives (ou les événements), appelé « frame of discernment », et dénoté  $\Theta$ . Dans notre problème de fermeture de boucle, « frame of discernment » est :

$$\Theta = \{C, I\} \quad (A.1)$$

où C représente la possibilité qu'une hypothèse de fermeture de boucle est correcte, et I représente la possibilité qu'elle est incorrecte. À la différence de la théorie des probabilités traditionnelle, la théorie de Dempster-Shafer nous permet d'assigner la probabilité aux ensembles de possibilités ; en particulier, la probabilité peut être assignée à n'importe quel membre de l'ensemble de puissance de  $\Theta$ ,  $2^\Theta$ .

Dans notre problème, c'est

$$2^\Theta = \{\emptyset, C, I, \{C, I\}\} \quad (A.2)$$

La théorie de Dempster-Shafer définit un « assignement de base de probabilité » dénoté BBA. Pour un « frame of discernment »  $\Theta$ , le BBA  $m : 2^\Theta \rightarrow [0, 1]$  est une fonction qui satisfait les conditions suivantes :

$$m(\emptyset) = 0 \quad (A.3)$$

$$\sum m(A) = 1 \quad (A.4)$$

L'équation A.3 indique qu'aucune croyance n'existe dans le  $\emptyset$ . L'équation A.4 spécifie la convention que la somme de croyance totale (de toutes les possibilités) devrait être un. Ici, la quantité  $m(A)$  est la masse assignée à A ; c'est la croyance mesurée exactement en A - mais pas toute la croyance mesurée en A. Toute la croyance en A est la somme de toute la croyance mesurée exactement en A et ses sous-ensembles, c.-à-d.

$$\text{Bel}(A) = \sum m(B), \text{ telle que } B \text{ est inclus dans } A. \quad (A.5)$$

La plausibilité de A est définie par :

$$\text{Pl}(A) = 1 - \text{Bel}(\bar{A}) \quad (A.6)$$

La plausibilité est la quantité de preuves qui ne soutienne pas la négation de A. La probabilité d'une possibilité particulière, dans le sens classique, est liée en-dessous par la croyance et en haut par la plausibilité [47], c.-à-d. :

$$\text{Bel}(A) \leq P(A) \leq \text{Pl}(A) \quad (\text{A.7})$$

Pour notre petite application, nous sommes principalement concernés par les probabilités de base  $m(C)$ ,  $m(I)$  et  $m(\{C, I\})$  – la preuve soutenant une hypothèse de fermeture de boucle, la preuve qui la réfute, et l'ignorance, respectivement.

Il est utile de pouvoir combiner des fonctions multiples de croyance. Par exemple, dans notre application, chaque mur que nous mesurons nous fournit la preuve que nous employons pour construire une fonction de croyance. Nous devons combiner les preuves fournies par de telles multiples mesures afin de maintenir une croyance « globale » de l'exactitude d'une hypothèse. Pour faciliter ceci, la théorie de Dempster-Shafer emploie la règle de combinaison de Dempster. Si on a deux BBA  $m_1$  et  $m_2$ , on peut les combinées par :

$$m_1 \oplus m_2(A) = \frac{\sum_{B, C \in 2^\Theta: B \cap C = A} m_1(B)m_2(C)}{1 - \sum_{B, C \in 2^\Theta: B \cap C = \emptyset} m_1(B)m_2(C)} \quad (\text{A.8})$$

Sauf si  $A = \emptyset$ , pour lequel :

$$m_1 \oplus m_2(\emptyset) = 0 \quad (\text{A.9})$$

Dans notre application, ceci se traduit aux règles de combinaison suivantes :

$$m_1 \oplus m_2(C) = \frac{m_1(C)m_2(C) + m_1(C)m_2(\{C, I\}) + m_1(\{C, I\})m_2(C)}{1 - m_1(C)m_2(I) - m_1(I)m_2(C)} \quad (\text{A.10})$$

$$m_1 \oplus m_2(I) = \frac{m_1(I)m_2(I) + m_1(I)m_2(\{C, I\}) + m_1(\{C, I\})m_2(I)}{1 - m_1(C)m_2(I) - m_1(I)m_2(C)} \quad (\text{A.11})$$

# ANNEXE B

## LA FONCTION DE CROYANCE POUR LES ENSEMBLES DES MESURES

Dans la section 3.3, nous avons présenté une approche de raisonnement par évidence pour le problème de « fermeture de boucle » pendant la phase de la cartographie (mapping) de base de notre algorithme de cartographie topologique. Cette approche est basée sur la théorie de preuve de Dempster-Shafer [49] ; pour une brève introduction à la théorie de Dempster-Shafer, voir l'annexe A.

On suppose que nous avons construit une hypothèse  $H_0 = \{\text{deux sommets dans notre carte sont identiques}\}$ . L'ensemble des mesures d'un mur est dénoté  $L$  et on écrit :

$$L = \{l_0, l_1, \dots, l_n\} \quad (\text{B.1})$$

On dénote la distribution d'incertitude liée à la mesure  $l_j$ , pris au début au sommet  $v_i$  et à la fin au sommet  $v_j$ , comme  $U_{v_i, v_j}$ . Cette incertitude est déterminée par le modèle d'erreur d'odométrie du robot. Typiquement, l'écart type de la distribution augmente avec la longueur d'une mesure. L'accumulation des preuves basée sur des ensembles de mesures n'assume aucune distribution particulière, tant que l'information de covariance est disponible.

Nous avons assumé la disponibilité d'un modèle raisonnablement précis de l'erreur provoquée par l'odométrie du robot. Notre fonction de croyance est conçue en adoptant une approche d'analyse statistique en utilisant ce modèle d'erreur.

## B.1 Propriétés de fonction de croyance

N'importe quelle fonction de croyance que nous créons devrait satisfaire certaines propriétés intuitives :

1.  $m_L(C)$  (la croyance que  $H_0$  est correct) devrait varier inversement avec la variance de  $L$  - un ensemble de mesures considérablement variées fournit peu de preuves que toutes les mesures sont du même mur.

2.  $m_L(I)$  (croyance que  $H_0$  est incorrecte) devrait varier de la même façon que la variance de  $L$  ; si la variance de  $L$  est zéro,  $m_L(I)$  devrait être zéro (un ensemble de mesures identiques ne devrait fournir aucune preuve contre  $H_0$ ).

3. Un seul ensemble de mesures peut réfuter  $H_0$ , mais un seul ensemble de mesures ne devrait pas généralement être suffisant pour prouver  $H_0$ .

## B.2 La fonction de croyance primaire

On définit la fonction de croyance pour un ensemble de mesures  $L$  qui, selon  $H_0$ , sont (les mesures) du même mur :

$$m_L(C) = \Phi(L)\eta_L \quad (\text{B.2})$$

$$m_L(I) = (1 - \Phi(L))\eta_L \quad (\text{B.3})$$

Ici,  $\Phi : L \rightarrow [0, 1]$ . Nous discuterons la fonction  $\Phi$  dans la section B.3.

La valeur  $\eta_L \in [0, 1]$  représente « la probabilité du contenu d'information » [56] de  $\Phi(L)$ , et elle est calculé comme suit :

$$\eta_L = 1 + \frac{\Phi(L) \log(\Phi(L)) + (1 - \Phi(L)) \log(1 - \Phi(L))}{\log(2)} \quad (\text{B.4})$$

La probabilité du contenu d'information  $\Phi(L)$  mesure essentiellement la « distance » de  $\Phi(L)$  de la distribution uniforme. Ainsi, ~~é~~ pour un  $\Phi(L)$  avec une entropie élevée (moins d'information), nous affirons plus d' « ignorance, » et pour un  $\Phi(L)$  avec une basse entropie (plus d'information), nous affirons moins d'ignorance.

Quand on traverse l'environnement après la construction de l'hypothèse, on prend de nouvelles mesures et on les utilise pour le calcul de la croyance selon les équations B.2 et B.3. Les valeurs obtenues sont combinées avec la croyance globale  $m$  de l'hypothèse  $H_0$  en utilisant la règle de combinaison de Dempster (voir la section 3.3).

### B.2.1 Valeurs initiales

Quand on construit une hypothèse, nous devons initialiser la croyance globale dans l'hypothèse. Une méthode simple serait d'assumer l'ignorance complète (c.-à-d.  $m(\{C, I\}) = 1$ ). Cependant, ce n'est pas vraiment le cas - nous avons une certaine croyance en l'exactitude de l'hypothèse, ou nous ne l'aurions pas faite du tout.

Rappelons cela géométriquement, aucunes deux caractéristiques du même type et orientation peut être dans un rayon  $r^*$  (voir la section 3.2). Ceci veut dire:

$$m_L(C) = \int_{x_0-r^*}^{x_0+r^*} \int_{y_0-\sqrt{r^*-(x-x_0)^2}}^{y_0+\sqrt{r^*-(x-x_0)^2}} U_{v_0, v_h}(x, y) d\mathbf{A} \quad (\text{B.5})$$

Tel que  $v_0$  est le sommet de début et

$$H_0 \equiv v_0 = v_h \quad (\text{B.6})$$

En d'autres termes, selon notre modèle d'erreur, la valeur initiale de  $m(C)$  représente la probabilité que  $|v_0 - v_h| \leq r^*$  (dans ce cas ils doivent représenter la même caractéristique). On initialise  $m(I)$  à zéro, cependant ; au moment où nous construisons  $H_0$ , nous n'avons aucune base pour assigner la masse de croyance à la possibilité que  $H_0$  soit incorrect.

### **B.2.2 L'arrêt**

Nous avons décrit une fonction de croyance pour déterminer la preuve donnée pour une hypothèse  $H_0$  par un ensemble de mesures qui, selon  $H_0$ , viennent du même mur. Quand est-il sûr que notre robot doit accepter ou rejeter  $H_0$  ? Une vraie approche d'« hypothèse multiple » pourrait maintenir un ensemble de toutes les hypothèses et leurs fonctions de croyance associées indéfiniment, et donc la décision est l'hypothèse correcte. Nous permettons à des hypothèses multiples de coexister, mais nous essayons également de prendre des décisions au sujet de leur exactitude.

Afin de faire ceci, nous choisissons un seuil  $\beta$  de la valeur de croyance. Si  $m(C)$  dépasse  $\beta$ , nous acceptons  $H_0$  ; si  $m(I)$  dépasse  $\beta$ , nous rejetons  $H_0$ . Ceci implique que pour accepter une hypothèse,  $Pl(I) < 1 - \beta$ .

Un choix raisonnable pour  $\beta$  est la limite de confiance  $\alpha$ . En fait, nous pouvons penser que  $\beta$  définit les limites de confiance sur notre croyance. Bien que la croyance ne soit pas vraiment équivalente à une probabilité classique que  $H_0$  est correct, notre croyance en  $H_0$  peut être interprétée comme la limite inférieure sur la probabilité qu'elle est ( $H_0$ ) correcte, pour les mesures que nous avons décrites. Voir l'annexe B pour plus d'information.

Puisque  $m(C)$  est une limite inférieure sur la probabilité que  $H_0$  est l'hypothèse correcte, mettant  $\beta \geq \alpha$  maintient la condition que toutes les « décisions » de cartographie soient faites avec une confiance d'au moins  $\alpha$ .

### B.2.3 Limitations

Malheureusement, notre fonction de croyance ne peut pas satisfaire parfaitement la troisième propriété indiquée dans la section B.1 (qu'un seul ensemble de mesures ne devrait pas être suffisant pour prouver  $H_0$ ). Nous pourrions rencontrer un mur jamais-vu- avant de la même longueur comme un mur que nous avons précédemment observé et avec précision. Dans ce cas-ci, notre fonction de croyance assigne une croyance forte à l'appui de  $H_0$ . Une telle situation est une faiblesse inhérente de l'approche - aucun robot ne peut savoir ce qu'il rencontrera dans un territoire encore inconnu. La fermeture de la boucle en mondes auto-similaires est possible avec nos méthodes mais reste encore difficile.

Il y a des approches que nous pourrions adopter pour remonter le risque des mondes auto-similaires, mais on doit payer le coût d'efficacité dans les environnements plus simples. En particulier, il pourrait être utile de développer une « mesure de particularités » pour des chemins entre les sommets : les chemins uniques devraient fournir une preuve plus forte. Les chemins « non-distinguant » est la difficulté primaire avec les environnements auto-similaires ; la décrémentation de  $m_L$  pour de tels chemins nous permettrait de chercher après eux, jusqu'à ce que nous rencontrions une disparité structurale ou différencier la longueur du trajet.

Naturellement, nous ne pouvons pas généralement savoir si un chemin est distingué sans connaissance antérieure du monde. Une approximation simple, que nous avons mise en application expérimentalement, est de compter le nombre de bords déjà dans la carte qui sont semblables à une mesure nouvellement prise. Si ce nombre est grand,  $m_L$  est décrémenté par conséquence. Dans beaucoup de cas cette méthode sert à aider à refuser des hypothèses incorrectes en mondes extrêmement autosimilaires (comme dans la Fig 5.4). Malheureusement, le pessimisme qui est conçu pour défaire des hypothèses incorrectes rend souvent les hypothèses correctes difficiles à prouver d'une façon efficace. En plus, cette méthode fournit peu d'avantage dans les environnements compliqués qui sont autosimilaires à grande échelle.

En pratique, peu de situations réelles existent qui sont assez autosymétriques. Pour cette raison, nous ne le présentons pas en tant que notre fonction primaire de croyance ; dans la grande majorité de cas, la fonction décrite dans la section B.2 se comporte bien.

### B.3 $\Phi$ -Fonction

Les équations B.2 et B.3 se rapportent à une fonction  $\Phi : L \rightarrow [0, 1]$  qui renvoie une valeur indiquant que la probabilité d'obtenir  $L$  sachant que tout les  $l_i \in L$  sont prises du même mur. Généralement les  $\Phi$ -fonctions dépendent du modèle d'erreur d'odométrie du robot. Nous supposons que les informations de covariance sur l'incertitude d'une mesure sont disponibles ou peuvent être calculées.

D'abord nous adressons le cas univariable, dans lequel suivre-mur est un mouvement « à ligne directe » sans la correction verticale. Nous prolongeons alors cette approche au cas à deux variables dans lequel le robot fait des corrections par rapport au mur pendant qu'il le traverse.

Sans perte de généralisation, nous supposons que  $|L|= 2$  dans toute cette discussion ; en particulier,  $L= \{l_j, l_k\}$ . Les résultats peuvent facilement être prolongés au cas pour lequel  $|L| = n \geq 2$ .

#### B.3.1 Erreur univariable

On suppose que notre modèle d'erreur est tel que  $U_{vi, vj}(x)$  et  $U_{vi, vk}(x)$  suivent des distributions univariées avec les écarts type  $\sigma_j$  et  $\sigma_k$ , respectivement. Sous l'hypothèse  $H_0$ , notre « meilleure évaluation » de la longueur vraie du mur à partir duquel nous avons obtenu  $\hat{l}$  de  $l_j$  et de  $l_k$ , est calculée comme suit :

$$\hat{l} = l_j + \frac{\sigma_j^2(l_k - l_j)}{\sigma_j^2 + \sigma_k^2} \quad (\text{B.7})$$

Pour  $\hat{\ell}$ , nous obtenons une distribution d'incertitude de position  $\hat{U}$  selon notre modèle d'erreur - non estimé à partir des échantillons en L - avec le  $\hat{\sigma}$  d'écart type.

Les z-points représentent la déviation d'un échantillon de la moyenne prévue, exprimée en unités de l'écart type de la distribution fondamentale. Dans notre cas, nous calculons la somme du  $z^2$  de nos échantillons pour la distribution  $\hat{U}$ , c.-à-d.

$$z_{j,k}^2 = \sum_{i \in \{j,k\}} \frac{(\ell_i - \hat{\ell})^2}{\hat{\sigma}^2} \quad (\text{B.8})$$

On note que  $z_{j,k}^2$  suit une distribution  $\chi^2$  avec deux degrés de liberté, équivalente à une distribution gamma avec  $\theta \equiv 2$  et  $\alpha \equiv 1$ , c.-à-d.

$$P(y) = \frac{e^{-y/2}}{2} \quad (\text{B.9})$$

Dans le cas général pour lequel  $|L| = n$ ,  $z^2$  suit une distribution  $\chi^2$  avec n degrés de liberté. Ce ci parce que nous estimons seulement la moyenne ( $\hat{\ell}$ ) à partir des échantillons en L (nous n'employons pas les échantillons pour estimer  $\hat{\sigma}$ ). Une distribution  $\chi^2$  avec n degrés de liberté est équivalente à une distribution gamma avec  $\theta \equiv 2$  et  $\alpha \equiv n/2$ , c.-à-d.

$$P(y) = \frac{y^{n/2-1} e^{-y/2}}{2^{n/2} \Gamma(n/2)} \quad (\text{B.10})$$

Nous employons cette statistique pour estimer le degré d'appui fourni par L pour  $H_0$ . En d'autres termes,

$$\Phi(L) = \int_{z_{j,k}^2}^{\infty} P(y) dy \quad (\text{B.11})$$

### B.3.2 Erreurs bivariées

Cette approche est facilement prolongée au cas bivariées (c.-à-d., le cas dans lequel « suivre – mur » implique un mouvement et une erreur bidimensionnels). Supposons que nous avons une paire de mesures (vecteur)  $L = \{l_j, l_k\}$ , et que  $U_{vi, vj}(x, y)$  et  $U_{vi, vk}(x, y)$  suivent des distributions à deux variables avec  $C_j$  et  $C_k$  représentant les matrices de covariance respective selon le modèle d'erreur. Sous  $H_0$ , notre « meilleur estimation » de la véritable longueur de mur est maintenant

$$\hat{\ell} = \ell_j + C_j(C_j + C_k)^{-1}(\ell_k - \ell_j) \quad (\text{B.12})$$

pour laquelle on obtient une distribution d'incertitude de position  $\hat{U}$  selon notre modèle d'erreur, avec la matrice de covariance associée  $\hat{C}$ .

La distance de Mahalanobis est l'équivalent multidimensionnel des z-points. En utilisant  $\hat{C}$ , on calcule la somme des distances carrées de Mahalanobis entre nos échantillons et  $\hat{\ell}$ , c.-à-d. :

$$M_{j,k}^2 = \sum_{i \in \{j,k\}} (\ell_i - \hat{\ell})^T \hat{C}^{-1} (\ell_i - \hat{\ell}) \quad (\text{B.13})$$

$M_{j,k}^2$  suit également une distribution  $\chi^2$  de deux degrés de liberté (ou n degrés de liberté pour  $|L| = n$ ). Ainsi, le degré d'appui donné par  $L$  pour  $H_0$  est

$$\Phi(L) = \int_{M_{j,k}^2}^{\infty} P(y) dy \quad (\text{B.14})$$

# ANNEXE C

## SPECIFICATION DES COMPORTEMENTS

En chapitre 3, nous avons présenté un algorithme de cartographie topologique pour les robots limités qui ont défini ses cartes en termes de comportements. Les comportements primaires utilisés dans l'algorithme de cartographie étaient : DEPLACER-VERS-MUR, SUIVRE-MUR et SUIVRE-COULOIR. Ici, nous détaillons ces comportements pour les mondes rectilignes.

Nous définissons les comportements pour un robot avec trois capteurs à courte portée seulement: un capteur en avant, un capteur gauche et un capteur droite qui retournent les portées  $r_f$ ,  $r_l$  et  $r_r$ , respectivement.

Nos comportements sont développés en supposant que le robot n'éprouve aucune erreur de détection ou d'odométrie. Même avec cette prétention, les comportements sont dus tout à fait compliqué à cause du grand nombre de scénarios qui peuvent se produire même en mondes rectilignes. Dans les travaux futurs, les comportements seront prolongés pour traiter les erreurs de mouvement et de détection.

Les comportements prennent plusieurs paramètres (Tableau C.1). Puisqu'ils sont employés pendant la cartographie robotique, les comportements placent des « sommets » dans la carte d'un robot comme appropriée ; les types de sommets qui peuvent être placés sont décrits dans le Tableau C.2.

Les comportements sont décrits en utilisant les automates finis en lesquelles les états représentent « les actions primitives » et les transitions représentent les

conditions qui se produisent pendant l'exécution de ces actions. La condition nul  $\emptyset$  indique qu'une action se termine de son propre Accord (aucune condition anormale n'est détectée). Les différents types d'actions primitives desquelles nos comportements sont construits sont montrés dans le Tableau C.3.

Paramètre	Description
$r_0$	Offset.
$r_{min}$	Offset minimum.
$r_{max}$	Offset maximum.
$r_w$	Demi-largeur minimale d'un couloir.

**Table C.1: Paramètres des comportements. Pour plus d'examen de ces paramètres, voir la section 3.2.**

Type de sommet	Description
INT	Coin intérieur.
EXT	Coin extérieur.
ILL	(ignoré).
PORTAL	Un mouvement vers un espace ouvert (ex. en quittant un couloir).
SPLIT	Un mouvement vers un couloir.
TERM	Fin d'un couloir. (pas de continuation)

**Table C.2: Les types de sommet dans la carte. Dans certains cas, une direction est indiquée en plaçant un sommet (par exemple « SPLIT(L) »). Cette direction est relative à la référence courante du robot, et peut être convertie en référence globale de la carte pour un futur usage.**

Action	Argument	Description
A{F,B}	Sensor( $r_R, r_L$ )	Détecter le mur avec la capteur (en avant ou vers l'arrière)
M{F,B}	Distance	Se déplacer par la distance spécifique (en avant ou vers l'arrière)
T{L,R}	Rien	Tourner (gauche ou droit) 90 degrés.
AL	Rien	Aligner le robot à $r_0$ du mur du côté droit.
CN	Rien	Prendre le centre du couloir.
FM	Rien	Avancer, en maintenant le $r_0$ si suivre-mur ou en centre si en suivre-couloir
GOTO	Rien	Se déplacer dans la carte connue au prochain endroit encore inconnu
RET	Rien	Retourner une certaine valeur

**Table C.3 : Actions primitives qui sont employées pour construire les comportements de cartographie. Sauf FM, GOTO et RET, toutes les actions primitives renvoient la distance que le robot a déplacée du temps où l'action a été lancée.**

Pour la clarté et de la réutilisabilité, les comportements principaux ont été décomposés en composants. Chaque composant peut prendre des paramètres comme entrée, et peut renvoyer une certaine valeur ou ensemble de valeurs. C'est également vrai pour les « actions primitives » dont les comportements se composent ; typiquement, ces actions renvoient la distance ont voyagé pendant leur exécution.

Dans certains cas il est préférable de stocker les valeurs retournées par un certain composant ou action. Comme exemple, stocker la distance voyagée pendant l'exécution de la commande « déplacer en avant x mètres » dans la variable  $\delta$ , on écrit :

$$MF x \rightarrow \delta \quad (C.1)$$

Certains composants de nos comportements renvoient une liste de valeurs. Pour de telles occasions, il est commode de définir des opérateurs de liste. En y fait, le seul

opérateur que nous exigeons est l'opérateur LAST. Une liste de R valeurs, LAST (R) renvoie le dernier élément dans la liste.

## **C.1 EXPLORER**

Avant que nous définissions les « vrais » comportements, on définit un comportement de niveau élevé « EXPLORER ». Ce comportement exécute simplement l'algorithme de cartographie de base de la section 3.2. EXPLORER est spécifié dans le Tableau C.4.

## **C.2 DEPLACER-VERS-MUR**

Le comportement de DEPLACER-VERS-MUR est très simple : le robot avance jusqu'à ce qu'une de ses capteurs détecte un mur. Le robot tourne alors pour aligner sa capteur « SUIVRE-MUR » - typiquement sa capteur droite - avec ce mur.

Malgré cette simplicité, il est difficile de mettre en application un comportement DEPLACER-VERS-MUR robuste pour un robot avec des capacités de détection limitées. Ceci est parce qu'un petit obstacle ou un alignement initial malheureux peut faire rencontrer le robot à un objet sans le sentir d'abord. Nous n'abordons pas ce problème ici. Dans la pratique, il est habituellement sans importance - pas parce que le robot ne rencontre jamais une telle situation, mais parce qu'elle a peu d'effet sur le processus de cartographie (la cartographie ne commence pas jusqu'à ce qu'après un DEPLACER-VERS-MUR réussi ait été accompli).

Le comportement DEPLACER-VERS-MUR est détaillé dans le Tableau C.5.

## **C.3 SUIVRE-MUR**

Le comportement mur-suivant primaire est spécifié dans le Tableau C.6. Il doit traiter plusieurs cas ; le plus compliqué de ces derniers se produit quand un état de « discontinuité » est détecté. Cette circonstance pourrait mener au placement d'un coin extérieur, d'un portail ou - pour faire des choses plus ambiguës - de tous les deux ! Le comportement suivre - mur doit s'approcher des murs qui causent un état produisant un coin intérieur plus étroitement que de  $r_0$ , car ils peuvent s'ouvrir sur des couloirs de largeur moins que  $r_0$ . Une série de mouvements « des capteurs » est employée pour exécuter cet examen.

Une grande partie du travail difficile est manipulé par les composants de CAPTEUR et EXO, qui sont également employés par le comportement suivre-couloir. Le composant de CAPTEUR (Tableau C.9) examine un bout droit du mur d'une certaine longueur spécifique, manipulant toutes les ouvertures dans le mur convenablement. Le composant EXO (Tableau C.12) est responsable de manipuler des états de « discontinuité », déterminant s'ils constituent les coins ou les ouvertures extérieurs du couloir.

#### C.4 SUIVRE-COULOIR

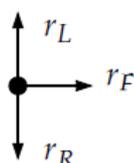
Le Tableau C.7 spécifie le comportement primaire suivre-couloir. Les conditions les plus compliquées en suivre-couloir, comme suivre-mur, se produisent quand des « discontinuité » sont détectées. En suivre-couloir, une discontinuité pourrait indiquer une intersection en T, un portail (extrémité-du-couloir qui s'ouvre dans un espace de non-couloir), une jonction à quatre voies ou un coin. Dans certaines circonstances, une jonction de plus de quatre couloirs peut se produire.

De nouveau, la majeure partie du travail est effectuée avec une série de mouvements CAPTEUR. En rencontrant une discontinuité, le robot détermine la nature de cette discontinuité, et puis revient au point initial de détection. Il capteur alors le long du mur vis-à-vis de la discontinuité. En conclusion, il capteur le mur vis-à-vis du point de détection.

Les impasses et les changements de largeur des couloirs sont traités tout comme les coins intérieurs dans suivre- mur. Le robot doit s'approcher près des coins afin de déterminer si des petites ouvertures existent.

#### C.5 Spécifications des composants

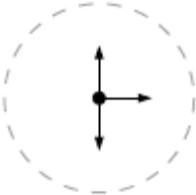
Dans les pages suivantes on va présenter les différentes spécifications pour chaque composant des comportements EXPLORER, DEPLACER-VERS-MUR, SUIVRE-MUR et de SUIVRE-COULOIR. Pour chaque composant, une description de son but est donnée, avec des conditions préalables, des post conditions, des arguments requis et des valeurs de retour. En conclusion, une série de situations dans lesquelles chaque composant est appliqué est donnée. Dans ces situations, le robot est représenté selon la convention montrée ici :



<b>EXPLORE</b>	
Implémentation	
Description	Explore un environnement rectiligne fermé en plaçant les sommets dans la carte.
Pré conditions	Rien
Post conditions	Rien (peut s'arrêter a n'importe quel moment par commande)
Arguments	$r_0$ , $r_{\min}$ , $r_{\max}$ , $r_w$ (Voir Table C.1)
Valeur de retour	Rien.

**Table C.4 Spécification de EXPLORE.**

<b>MTW(X)</b>	
Implémentation	
Description	Avancer jusqu'à ce qu'un capteur détecte un mur dans la portée $r_0$ . Tourner de sorte que la capteur dans la direction X soit alignée avec le mur.
Pré conditions	Rien
Post conditions	Le robot est de distance $r_0$ du mur sur sa droite.

Arguments	X : le coté du robot (Soit L ou R) qui doit être en face au mur.
Valeur de retour	Rien.
Situations	 <p>pas de mur dans la portée.</p>

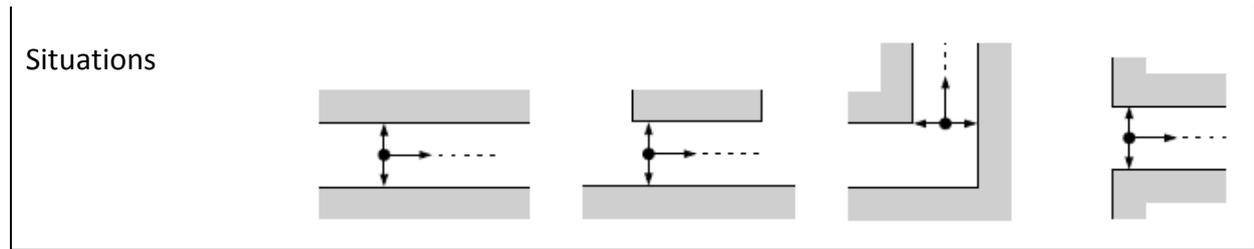
**Table C.5 DEPLACER-VERS-MUR.**

<b>WF</b>	
Implémentation	
Description	Comportement primaire suivre-mur. Le robot se déplace tout au long du mur dans un espace ouvert, maintenant une distance $r_0$ du mur. Les coins intérieurs et les coins extérieurs sont détectés, et des nœuds sont placés.
Pré conditions	Le robot doit pouvoir détecter un mur avec sa capteur droite ; en plus, ce doit être $r_L + r_r > 2r_0$ .
Post conditions	Rien (peut s'arrêter a n'importe quel moment par commande)
Arguments	Rien.

Valeur de retour	Rien.
Situations	

**Table C.6 SUIVRE-MUR.**

<b>HF</b>	
Implémentation	
Description	Comportement primaire suivre-couloir. Le robot se déplace tout au long des couloirs, en maintenant une distance égale des deux murs sur les deux cotés
Pré conditions	Le robot doit pouvoir détecter les deux mur sur ses deux cotés ;en plus, $r_L + r_r \leq 2r_0$ .
Post conditions	Rien (peut s'arrêter a n'importe quel moment par commande)
Arguments	Rien.
Valeur de retour	Rien.



**Table C.7 SUIVRE-COULOIR.**

<b>HFO(X)</b>	
Implémentation	
Description	Manipuler un état de « discontinuité » tout en SUIVRE-COULOIR.
Pré conditions	Le robot doit être placé à une discontinuité du côté X ( $r_x > r_{max}$ ).
Post conditions	Le robot a examiné l'ouverture, le mur vis-à-vis de l'ouverture et le mur directement devant sa position originale, et des nœuds sont ajoutés à sa carte comme appropriée. Le robot revient alors à SUIVRE -COULOIR.
Arguments	X: direction (Soit R or L) dans laquelle une condition d'une discontinuité a été détectée.
Valeur de retour	Rien.

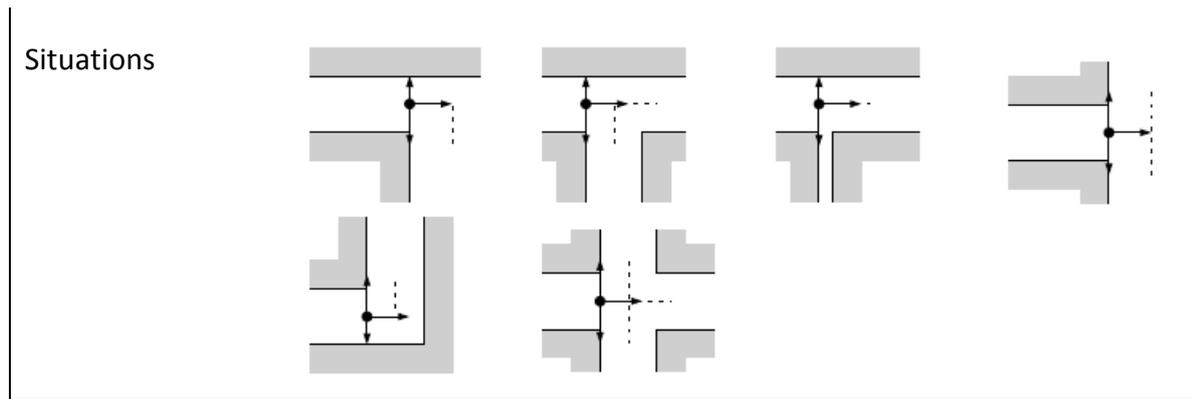


Table C.8 HFO(X).

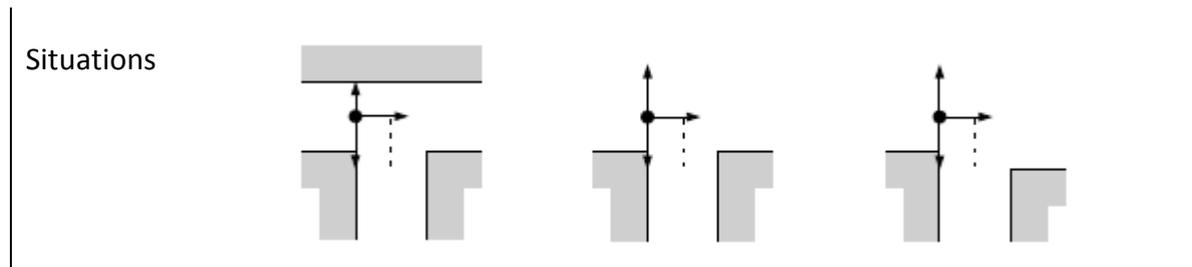
<b>PROBE(X,D)</b>	
Implémentation	
Description	Explore le mur dans la direction X sur une distance D. De multiples ouvertures peuvent être détectées et des nœuds sont placés.
Pré conditions	Le robot doit être placé à côté du mur sur son côté X .
Post conditions	Le robot a mis à jour sa carte avec tous les noeuds appropriés et s'est replacé à son point initial.
Arguments	X: direction (Soit R or L) D : longueur du mur.
Valeur de retour	Retourne une liste des nœuds placés au cours de l'exploration.

Situations



Table C.9 PROBE(X,D).

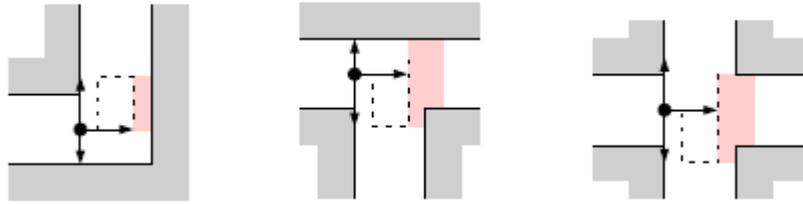
ENTER(X)	
Implémentation	
Description	Entre une « split » dans la direction X.
Pré conditions	Le robot doit être placé à la fin d'un couloir (suivre-couloir) ou à une discontinuité (suivre-mur) ; c.-à-d. $r_x > r_{max}$ . Le couloir ou le mur courant doit être relié, par l'intermédiaire d'un nœud de « split », à un couloir dans la direction X.
Post conditions	
Arguments	X: direction (Soit R or L)
Valeur de retour	Retourne $r_w$



**Table C.10 ENTER(X).**

<b>OPP(X,D)</b>	
Implémentation	
Description	Examine le mur opposé.
Pré conditions	Le robot doit être placé à la fin d'un couloir; c.-à-d. $r_x > r_{max}$ . Le couloir courant doit être relié, par l'intermédiaire d'un nœud de « split », à un couloir dans la direction X.
Post conditions	Le robot est placé à son emplacement original. Toutes les ouvertures le long du mur dans la direction vers l'avant ont été détectées et reliées (par l'intermédiaire de « split » ou des nœuds PORTAL, comme approprié) au couloir dans la direction X.
Arguments	X: direction (Soit R or L) D : Distance. Typiquement $2r_0$ .
Valeur de retour	Rien.

Situations



**Table C.11 OPP(X,D).**

<b>EXO(X,D,N)</b>	
Implémentation	
Description	Explore une seule ouverture.
Pré conditions	Le robot doit être placé au début d'une discontinuité sur le côté X.
Post conditions	Le robot a mis à jour sa carte avec tous les nœuds appropriés.
Arguments	X: direction (Soit R or L) D : Distance maximum. N : Type du nœud à placer.
Valeur de retour	Retourne un pair : le type de nœud placé et la distance voyagé.

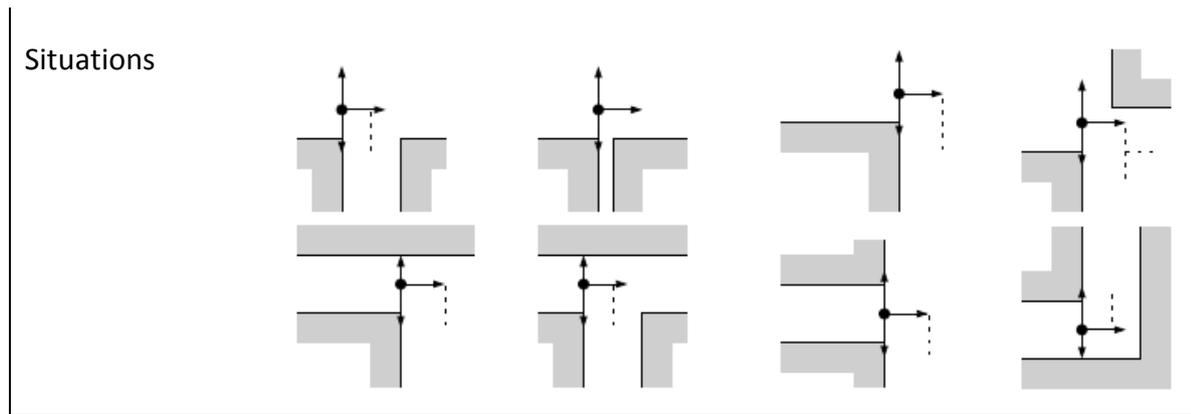
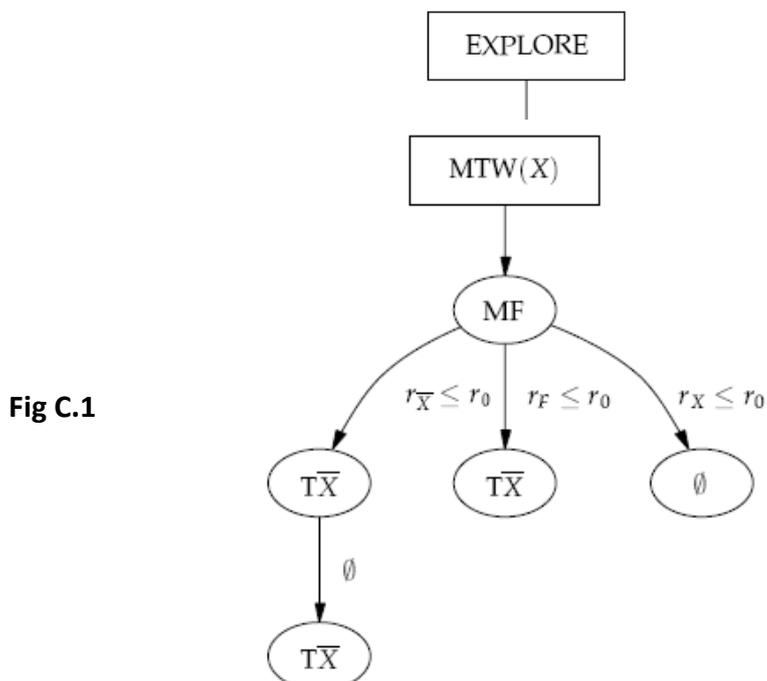


Table C.12 EXO(X,D,N).

### C.6 Réalisations des composants

Dans les pages suivantes sont les réalisations de chaque composant des comportements de cartographie. Ces réalisations sont présentées en tant que automate à état fini, construites utilisant les actions primitives décrites dans le Tableau C.3. Chaque nœud représente l'exécution d'une action primitive. La plupart des actions pourraient se terminer dans plusieurs conditions ; les bords entre les nœuds indiquent la condition de terminaison de l'action dans le nœud duquel ils commencent. Si aucune des conditions spécifiques n'est remplie, le  $\emptyset$  nul de condition spécifie la prochaine action pour prendre.



**L'implémentation de EXPLORER.**

**Fig C.2 L'implémentation de DEPLACER-VERS-MUR.**

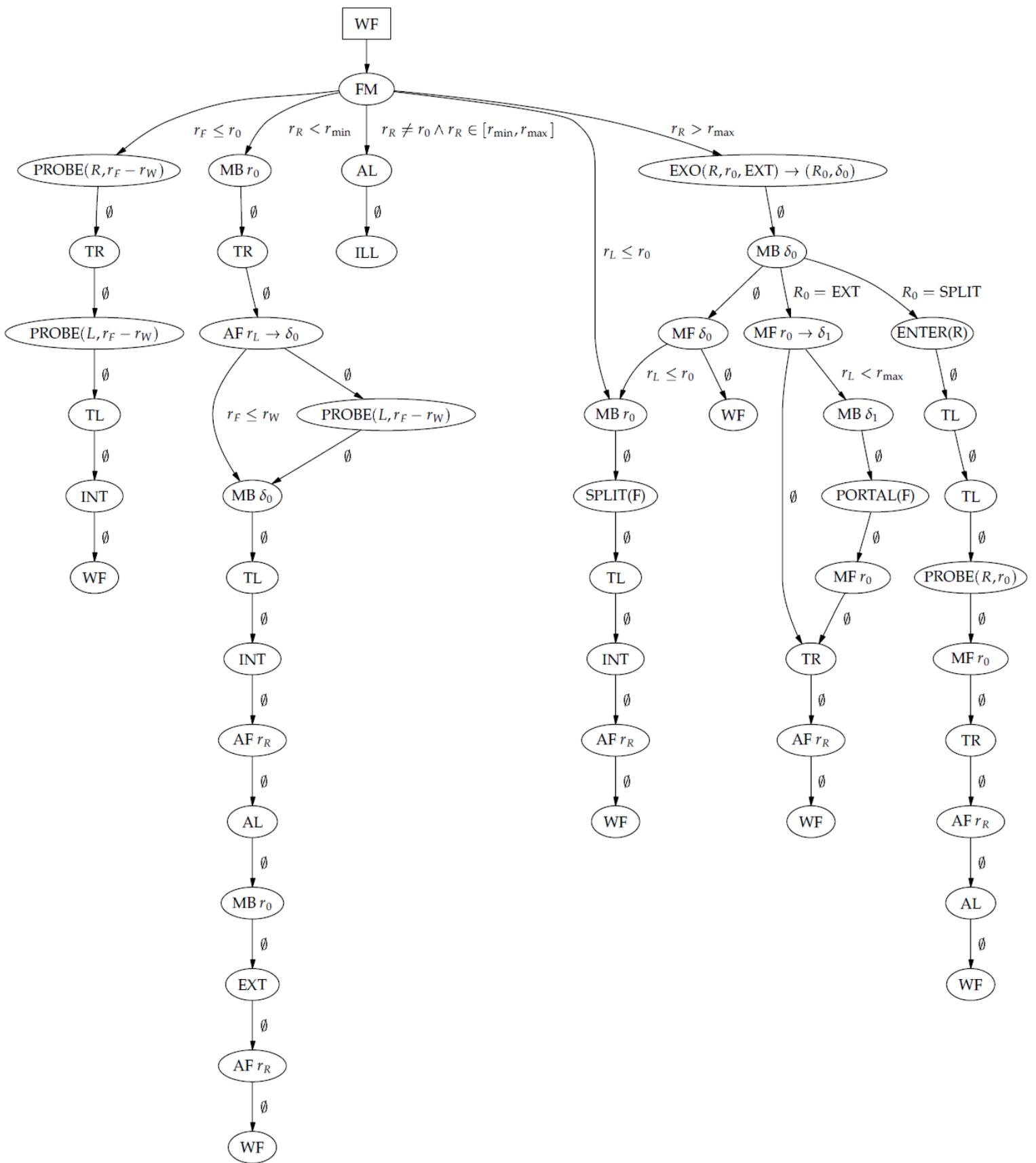


Fig C.3 L'implémentation SUIVRE-MUR.

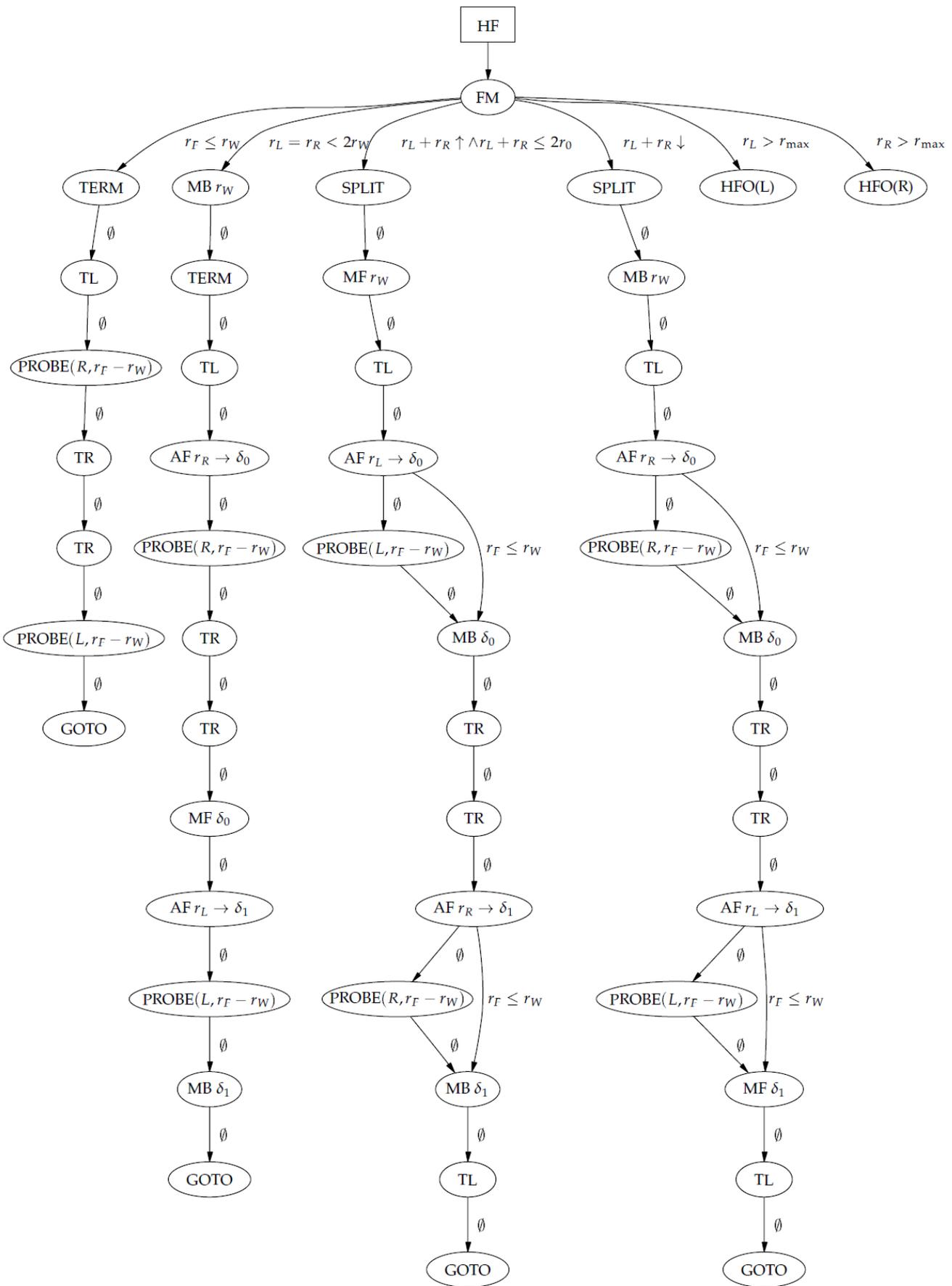


Fig C.4 L'implémentation de SUIVRE-COULOIR.

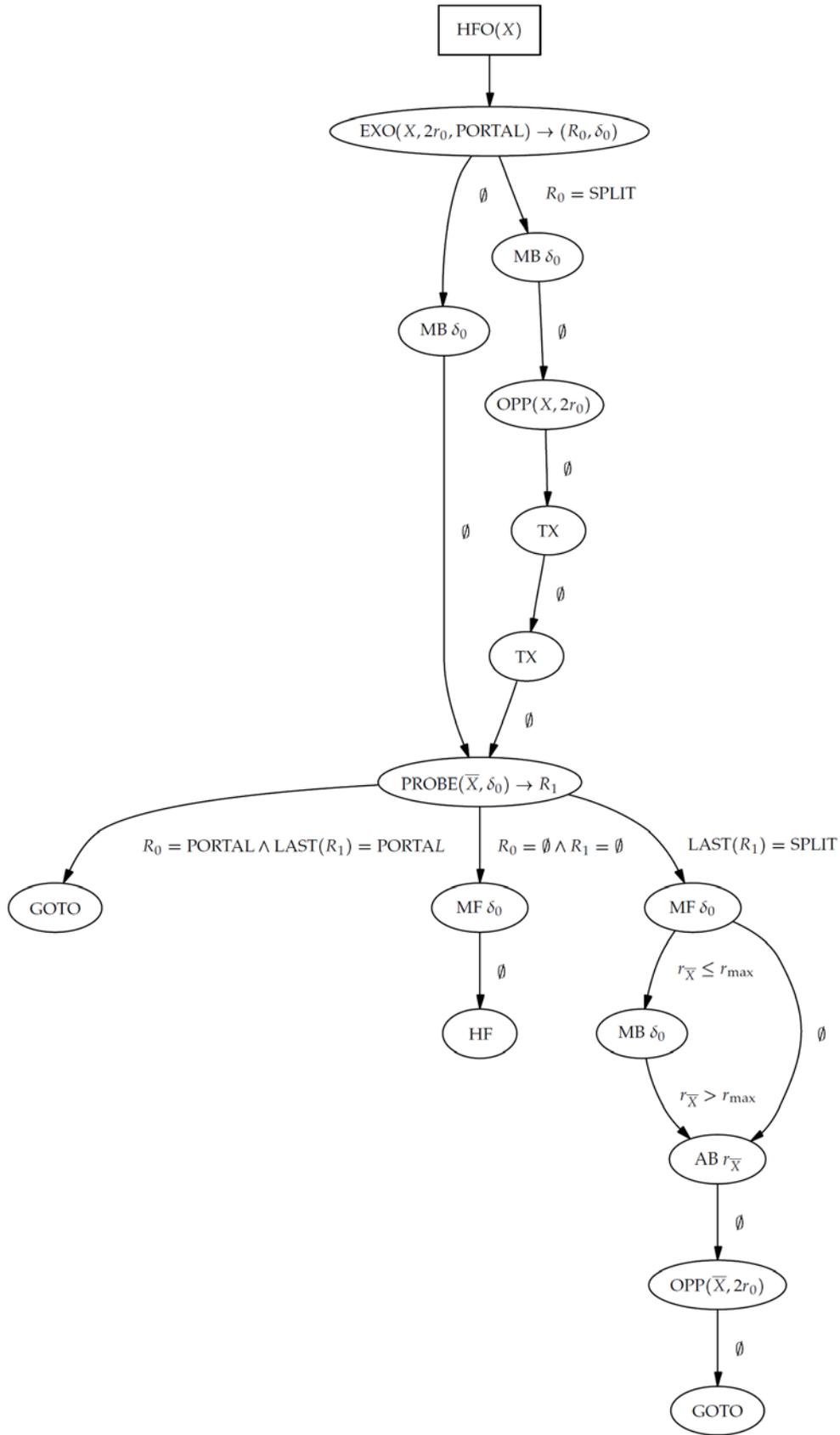


Fig C.5 L'implémentation du composant HFO.

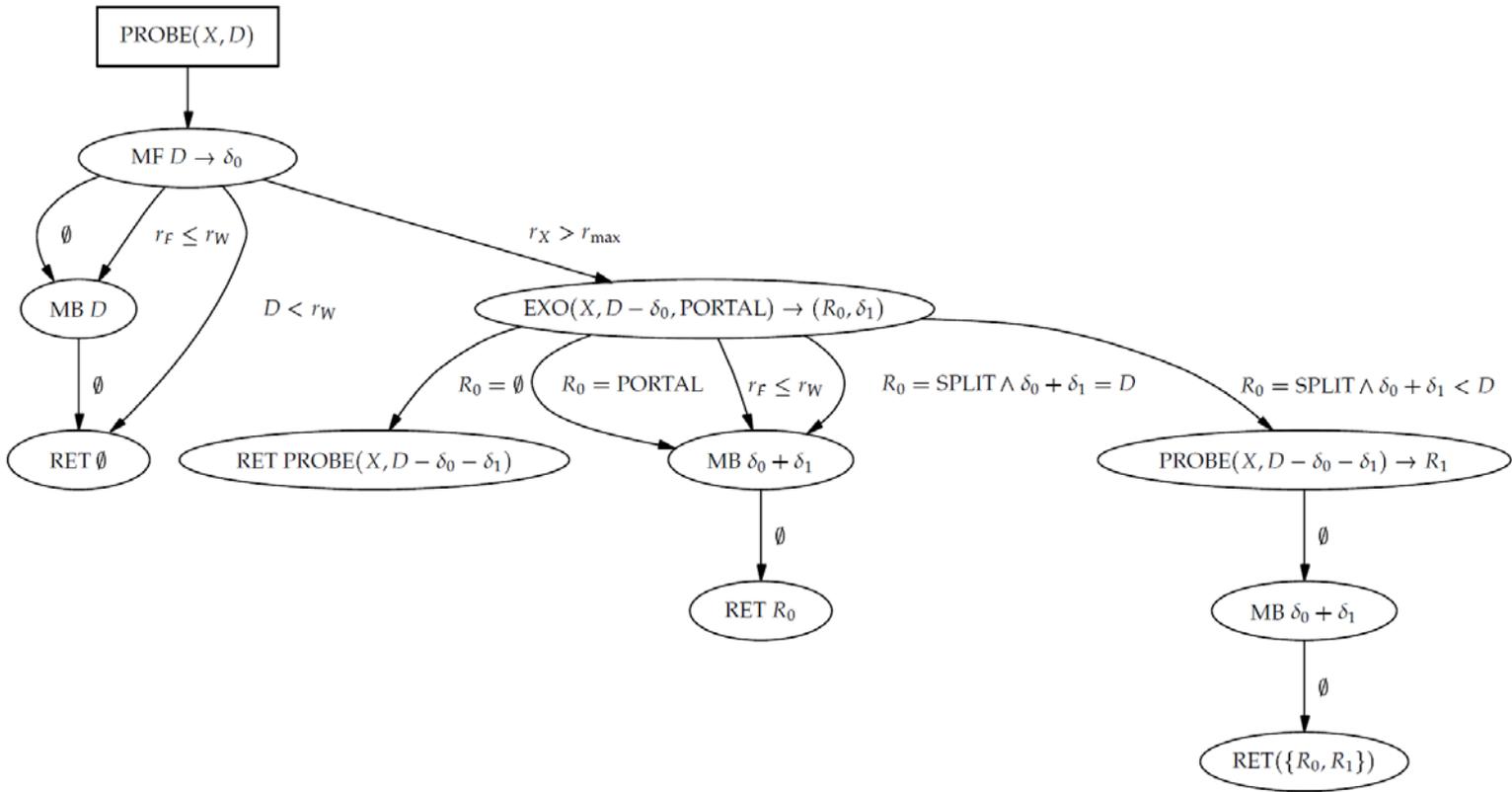


Fig C.6 L'implémentation de PROBE.

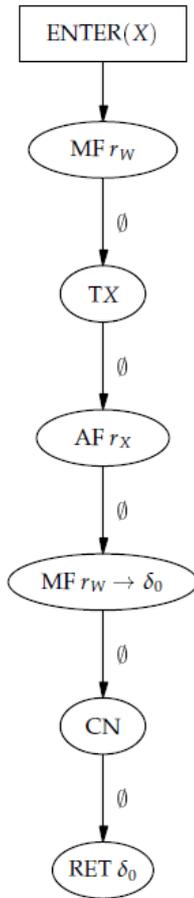


Fig C.7 L'implémentation du composant ENTER.

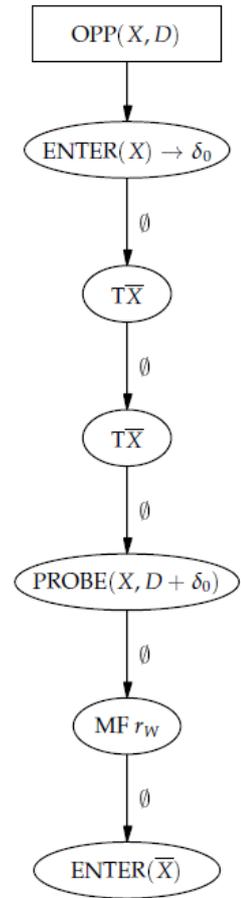


Fig C.8 L'implémentation du composant OPP.

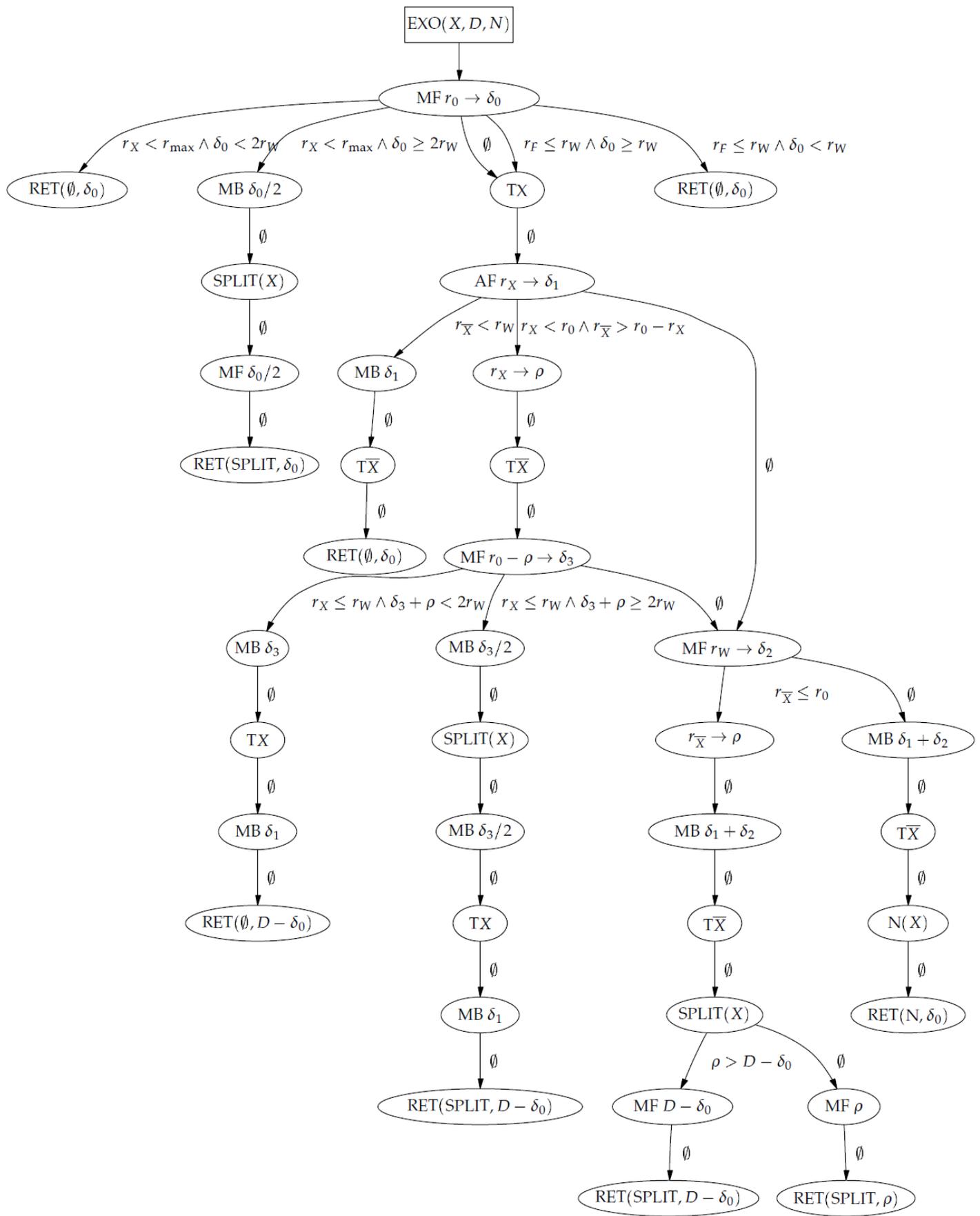


Fig C.9 L'implémentation du composant EXO.