

# Preuves de déconnexion

Antoine Amarilli

## Résumé

Les algorithmes de Probabilistic Roadmap (PRM) permettent de planifier le mouvement d'une façon efficace, mais ils sont malheureusement incapables d'identifier de façon sûre les cas où aucun chemin n'existe. La recherche de *preuves de déconnexion* vise à conclure à l'absence de chemin dans les cas où cette absence est manifeste ; à l'inverse de PRM, elle ne conclut pas nécessairement dans les cas où un chemin existe.

Nous décrivons ici une technique de preuve de déconnexion mise en œuvre par [1], qui s'intéresse à un cadre particulier où un robot tente de franchir une ouverture polygonale découpée dans un obstacle plan. Nous présenterons également les résultats expérimentaux, ainsi que les prolongements envisagés.

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Démarche</b>	<b>2</b>
2.1	Coupe . . . . .	2
2.2	Continuité de preuve . . . . .	3
2.2.1	Réduction . . . . .	3
2.2.2	Tesselation de la sphère . . . . .	3
2.3	Inclusion de polygones . . . . .	3
<b>3</b>	<b>Résultats expérimentaux</b>	<b>4</b>
<b>4</b>	<b>Conclusion et perspectives</b>	<b>5</b>

## 1 Introduction

Les algorithmes de type PRM sont une manière efficace de résoudre des problèmes de planification de mouvement. Leur fonctionnement consiste à générer en premier lieu une feuille de route (*roadmap*), qui sera ensuite utilisée pour répondre à des requêtes demandant de déterminer un chemin admissible pour aller d'un point de départ à une destination.

La génération de la feuille de route s'effectue en tirant des configurations au hasard, en déterminant si elles sont admissibles, et, si elles le sont, en les raccordant à des configurations proches à l'aide d'un algorithme local de planification de mouvement, l'espoir étant que la génération aléatoire des configurations permettra d'échantillonner efficacement la totalité de l'espace libre. Cette approche a le mérite de ne pas nécessiter de caractérisation exacte de l'espace des configurations. En revanche, elle a le défaut de ne pas permettre de conclure à l'absence d'un chemin pour un problème donné.

En effet, étant donné que l'échantillonnage est aléatoire, si aucun chemin n'est trouvé, on ne peut pas savoir si c'est parce que le nombre de configuration générés est insuffisant ou parce qu'il n'y a effectivement aucun chemin admissible. Par exemple, dans le cas de deux grandes régions séparées par un passage étroit où le robot passe de justesse, il est nécessaire d'essayer un grand nombre de configurations avant de tomber sur celle qui permet de faire le lien entre les deux composantes.

Certes, il est possible d'avoir des garanties probabilistes permettant de restreindre la probabilité d'erreur en fonction de certaines propriétés de l'espace et en fonction du nombre de configurations testées. Cependant, plutôt que de tester un très grand nombre de configurations pour réussir à limiter suffisamment le risque de ne pas trouver le chemin voulu, il semblerait plus satisfaisant de prouver à la place que ce chemin n'existe pas, surtout dans les cas où il est manifeste que le robot ne pourra pas passer.

## 2 Démarche

On s'intéresse, à des fins illustratives, au problème simplifié suivant. Un robot polyédral rigide se déplace dans un espace à trois dimensions, et cherche à traverser une ouverture polygonale découpée dans un obstacle plan infini. En d'autres termes, on choisit deux configurations arbitraires de part et d'autre de l'obstacle, et on cherche à déterminer si un chemin permet de les relier.

L'idée permettant d'aboutir à des preuves de déconnexion est la suivante. Pour tout  $\alpha \in ]0, 1[$ , s'il existe un chemin permettant au robot de passer par l'ouverture, alors, par continuité, il existe une position intermédiaire du robot sur ce chemin en laquelle une fraction  $\alpha$  du volume du robot se trouve d'un côté de l'ouverture et une fraction  $1 - \alpha$  de l'autre côté. Autrement dit, par contraposition, s'il existe un  $\alpha$  tel qu'aucune telle position intermédiaire existe, alors aucun tel chemin n'existe.

Remarquons qu'une orientation du robot suivant une certaine direction définit une unique position de ce dernier telle que son volume soit réparti de part et d'autre de l'obstacle suivant les proportions  $\alpha : 1 - \alpha$  (la méthode de calcul de cette position est décrite en section 2.1). En conséquence, pour un  $\alpha$  donné, si, pour toutes les directions de l'espace, la position correspondante du robot n'est pas admissible (ie. sa section suivant le plan de l'obstacle n'est pas contenue dans l'ouverture), alors il est impossible de franchir l'obstacle.

Il est bien entendu impossible d'énumérer la totalité des orientations possibles de l'espace (vu qu'il y en a une infinité). En conséquence, la méthode utilisée consiste à recouvrir la sphère des orientations par des morceaux correspondant à des preuves de l'impossibilité de faire passer le robot pour un certain voisinage autour d'une direction donnée. La méthode précise pour ce faire est décrite en section 2.2.

Pour une orientation donnée, il reste à déterminer si la position correspondante du robot tient ou non dans l'ouverture. Ce point est expliqué en section 2.3.

### 2.1 Coupe

On rappelle que dans cette section, on cherche, pour une direction donnée et une proportion  $\alpha$  donnée, à déterminer un plan ayant cette direction pour normale qui coupe le robot de sorte à répartir son volume suivant  $\alpha : 1 - \alpha$ . On triangule si nécessaire les faces du robot au préalable.

Dans ce contexte, souvenons-nous de l'algorithme standard pour calculer le volume du robot : on choisit un point de référence arbitraire, et on somme le volume signé des tétraèdres ayant pour sommet le point de référence et pour base un des triangles des faces du robot.

Cet algorithme s'étend sans grande difficulté au calcul du volume du robot qui se trouve d'un côté donné du plan de l'obstacle. On commence par trier tous les sommets du robot par coordonnées  $z$  croissantes, où on choisit comme axe  $\vec{z}$  la normale au plan de l'obstacle. Ensuite, le volume du robot d'un côté du plan est la

somme du volume signé des tétraèdres qui sont complètement d'un côté du robot plus un polynôme de degré 3 en  $z$  pour les tétraèdres qui sont coupés par le plan.

On en déduit sans peine un algorithme permettant, pour une proportion  $\alpha : 1 - \alpha$  donnée et une orientation du robot donnée, de déterminer la coordonnée  $z$  du plan coupant le robot suivant cette proportion (et donc la position du robot dans le cas où l'obstacle est fixe et où c'est le robot qui se déplace). En premier lieu, on procède par dichotomie pour trouver l'intervalle de  $z$  *sans sommets du robot* qui convient. Ensuite, on détermine la valeur exacte de  $z$  en résolvant le polynôme de degré 3 exprimant le volume en fonction de  $z$  dans cet intervalle.

## 2.2 Continuité de preuve

Dans cette section, on cherche à déterminer un moyen, pour une proportion  $\alpha : 1 - \alpha$  fixée, de tester toutes les orientations possibles du robot afin d'essayer de déterminer qu'aucune d'entre elles ne convient.

### 2.2.1 Réduction

L'idée de base nous permettant de réaliser cela est la suivante : au lieu de tester si le robot passe ou non, nous chercherons à déterminer si une version rétrécie (ou, plus précisément, érodée) du robot peut passer ou non. Si le petit robot peut passer, il faudra réessayer avec un robot plus proche de l'original. En revanche, si le petit robot ne passe pas, nous pourrions établir que le robot original ne peut pas non plus passer pour toutes les directions formant un angle suffisamment petit avec la direction testée.

Pour énoncer les choses de façon plus précise, on introduit le formalisme suivant. On désigne par  $R$  le robot, et, pour tout réel  $\epsilon > 0$ , par  $R^\epsilon$  l'ensemble des points de  $R$  qui sont à une distance  $\leq \epsilon$  de la frontière de  $R$ . (Il s'agit formellement de la différence de Minkowski de  $R$  avec la boule de rayon  $\epsilon$ .) On note  $d$  la direction d'étude. On note  $\Pi$  le plan de l'obstacle. Enfin, on note pour toute direction  $d'$   $S_{d'}$  la  $\alpha$ -section de  $R$  suivant  $\Pi$ , ie. le polygone qui est l'intersection de  $R$  et de  $\Pi$  dans la configuration où  $R$  est orienté suivant  $d'$  et placé de sorte à ce que  $\Pi$  répartisse le volume de  $R$  suivant la proportion  $\alpha : 1 - \alpha$ . On note de même  $S_{d'}^\epsilon$  la  $\alpha$ -section de  $R^\epsilon$ .

Le résultat-clé est le lemme suivant (lemme 4.1 de [1]) :

**Lemme.** Notons  $\delta$  le diamètre de la projection de  $R$  sur  $\Pi$ . Si  $S_d^\epsilon$  ne tient pas dans l'ouverture, alors pour toute direction  $d'$  formant avec  $d$  un angle  $< \frac{\epsilon}{\delta}$ ,  $S_{d'}$  ne tient pas non plus.

(Par "tenir dans l'ouverture", on parle d'inclusion à rotation et translation près. Formellement, pour  $G$  l'ouverture, un polygone  $X$  tient dans  $G$  s'il existe une rotation  $\rho$  et une translation  $\tau$  telles que  $(\tau \circ \rho)(X) \subset G$ .)

Ainsi, pour une direction  $d$  donnée, si  $S_d$  ne tient pas, alors nous pouvons éliminer un voisinage de  $d$  : il suffit de trouver un  $\epsilon$  tel que  $S_d^\epsilon$  ne tient pas non plus.

### 2.2.2 Tessellation de la sphère

Le lemme précédent suggère un algorithme simple permettant de tenter d'énumérer de façon efficace toutes les directions de la sphère pour conclure à l'impossibilité de placer le robot.

On divise d'abord la sphère en huit grandes régions que l'on essaiera de tester avec un  $\epsilon$  adapté. Pour une telle région, on prend la direction  $d$  qu'elle représente, on calcule  $\delta$ , et on choisit  $\epsilon$  de sorte à ce que  $\frac{\epsilon}{\delta}$  soit plus grand que la taille de la région (ie. le plus grand angle sous lequel la section est vue depuis l'origine). Trois cas sont possibles :

- Si  $S_d^\epsilon$  ne tient pas, la région est éliminée (impossible de placer dans cette section).
- Si  $S_d$  tient, la preuve échoue (on a un placement admissible du robot qui respecte la proportion  $\alpha : 1 - \alpha$ ).
- Si  $S_d^\epsilon$  tient mais pas  $S_d$ , alors  $\epsilon$  est trop grand pour conclure, et on subdivise la région en régions plus petites que l'on teste récursivement.

Un exemple de tessellation de la sphère est fourni en figure 1.

Évidemment, le nombre d'étapes de cet algorithme n'est pas borné a priori ; on ne peut pas savoir quelle finesse de  $\epsilon$  sera nécessaire pour conclure soit à l'impossibilité de placer (une preuve a été obtenue pour chaque direction), soit à la possibilité de placer (un placement acceptable a été trouvé). En conséquence, il est opportun de prévoir une borne minimale sur  $\epsilon$  en dessous de laquelle la recherche doit être abandonnée.

## 2.3 Inclusion de polygones

Dans cette section, nous étudions comment il est possible de déterminer si un polygone (l'ouverture) en contient un autre (la section du robot), modulo rotation et translation.

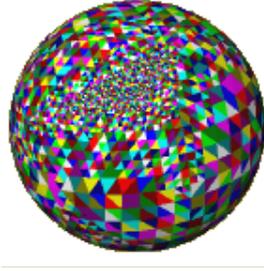


FIGURE 1 – Tessellation de la sphère.

La rotation est éliminée d’une manière analogue à celle de la section 2.2. On énumère toutes les rotations possibles en prouvant pour chaque rotation qu’un voisinage autour d’elle ne convient pas, à l’aide d’une version rétrécie de la section. Là encore, on commence par un découpage grossier, en raffinant progressivement si on ne sait pas répondre, et en échouant au-delà d’une certaine précision maximale.

Reste la question de la translation, qui nous ramène au problème général d’inclusion de polygones. Dans le cas d’une ouverture convexe, on peut prendre l’enveloppe convexe de la section, et se ramener à deux polygones convexes, cas pour lequel un algorithme linéaire existe (fournissant une preuve de taille constante).

Dans le cas général, on peut essayer de décomposer l’ouverture en polygones convexes disjoints, ou mettre en œuvre certaines pistes plus générales suggérées par l’article (p.6).

### 3 Résultats expérimentaux

La méthode a été testée dans le cas d’une ouverture carrée pour trois formes de robot (cf. figure 2) : un parallélépipède, un lambda majuscule, et un “serpent” (squelette polygonal de cinq segments). Les tests ont été effectués en faisant varier des paramètres de la forme du robot (taille du parallélépipède, écartement des jambes du lambda, longueur des segments du serpent), pour évoluer d’une situation où le robot passe clairement à une situation où le robot ne peut clairement pas passer.

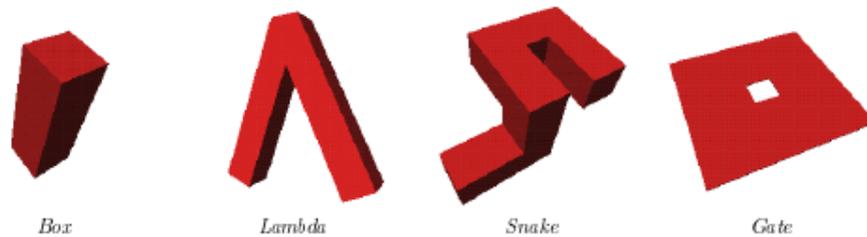


FIGURE 2 – Forme des robots et de l’ouverture

En plus de la figure 1, qui montre que la valeur de  $\epsilon$  requise dépend de l’orientation, des graphes sont fournis qui montrent l’évolution du nombre de triangles (ie. régions de la sphère) nécessaires pour conclure en fonction du paramètre clé pour la forme étudiée (cf. figure 3).

Pour le parallélépipède, les résultats sont intuitifs. Pour une forme qui peut clairement passer, peu de triangles sont nécessaires pour découvrir qu’on ne pourra pas conclure. En augmentant la taille, le nombre de triangles croît. Un saut marque le passage d’une forme qui passe de justesse à une forme qui ne passe pas de justesse. Le nombre de triangles nécessaires se met alors à décroître alors qu’il est de plus en plus évident que la forme ne passe pas.

Pour le serpent, les résultats sont plus surprenants. En particulier, pour un robot de grande taille, le nombre de triangles requis se met à croître à nouveau. En fait, cela peut s’expliquer par le fait que, dans le cas d’un objet grand, le diamètre  $\delta$  de la projection est grand, et donc il est nécessaire d’utiliser des valeurs fines de  $\epsilon$ . Cela suggère qu’il y a sans doute moyen d’améliorer la borne du lemme 4.1.

Notons que les graphes ne montrent pas le cas où l’algorithme ne conclut pas. En effet, celui-ci ne s’est jamais présenté au cours des tests.

## 4 Conclusion et perspectives

L'article suggère de nombreux développements possibles, et ce d'autant plus que les résultats qu'il présente sont préliminaires.

Dans un premier temps, un certain nombre de choses restent, de l'aveu même des auteurs, à implémenter. En pratique, le calcul de la différence de Minkowski est effectué par des méthodes ad hoc selon le robot testé. L'optimisation du code n'a pas encore été effectuée. Du reste, la méthode d'estimation du voisinage dans lequel une preuve est valide est perfectible.

La méthode présentée se limite à un cadre finalement assez restrictif (surtout pour ce qui est de l'obstacle). L'article envisage une extension immédiate au cas dans lequel l'obstacle est d'épaisseur finie (et non plus nulle). Il est également question d'une possibilité, dans un cadre de planification de mouvement plus général, de couper les espaces étroits suivant des plans arbitraires pour se ramener au cadre étudié.

On remarque aussi qu'un planificateur PRM et un prouveur de déconnexion sont, en un certain sens, complémentaires. Le premier tente de trouver un chemin mais ne remarquera pas de façon sûre qu'il n'en existe pas, le second tente de prouver qu'il n'y a pas de chemin mais ne remarquera pas de façon sûre qu'il en existe un. Qui plus est, les positions du robot qui font échouer la preuve de déconnexion peuvent constituer de bons *milestones* à indiquer au planificateur PRM pour l'aider à trouver un chemin.

Enfin, on remarque que l'utilisation d'un prouveur de déconnexion a le mérite de fournir une preuve qui peut fournir une explication quant à la déconnexion, au contraire d'un planificateur PRM qui se contenterait d'échouer sans fournir de motif. Cela pourrait être utile pour savoir comment modifier le robot ou l'environnement afin de rendre le passage possible.

## Références

- [1] J. Basch, L.J. Guibas, D. Hsu, and A.T. Nguyen. Disconnection proofs for motion planning. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pages 1765–1772, 2001.

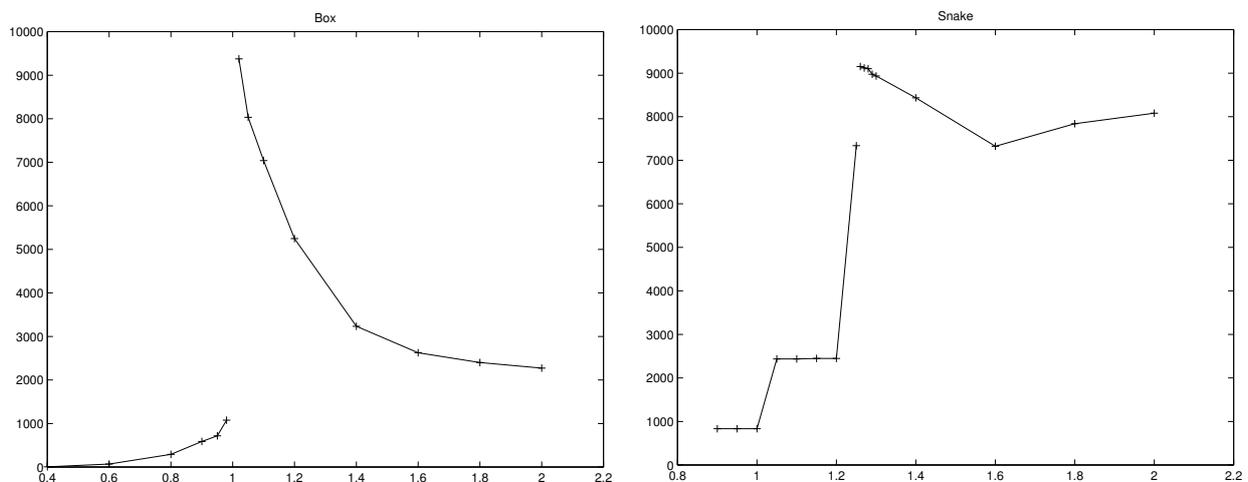


FIGURE 3 – Évolution du nombre de triangles pour conclure en fonction de la taille du robot.