Planification de tâches de manipulation par des méthodes probabilistes

Anis Sahbani*

Directeur(s) de thèse : Thierry Siméon

Laboratoire d'accueil :
LAAS-CNRS
7, avenue du Colonel-Roche
31077 Toulouse Cedex - France

Etablissement d'inscription :
Université Paul Sabatier

Résumé

Le problème de la planification de tâches de manipulation consiste non seulement à planifier les mouvements sans collision pour un robot, mais également le séquencement des différentes sous-tâches de prise et de pose d'objets. Les méthodes probabilistes sont utilisées afin de "casser" la complexité exponentielle de la planification du mouvement.

Après avoir présenté un bref état de l'art de ce domaine, nous détaillons la modélisation géométrique retenue de ce problème. Nous présentons également les algorithmes en cours de développement pour la conception de planificateurs appliqués aux problèmes de manutention dans des installations industrielles (cadre du projet Européen Molog)

Mots-clés

Planification de mouvement, méthodes probabilistes, tâches de manipulation, graphe de manipulation.

1 Introduction

La planification de mouvement a des applications dans plusieurs domaines tels que la robotique, la conception assistée par ordinateur, la réalité virtuelle, ou encore dans le domaine de la chimie moléculaire. Dans sa version la plus simple, le problème est intrinsèquement difficile, sa complexité croit exponentiellement avec le nombre de degrés de liberté du robot. Traditionnellement les algorithmes proposés dans la littérature conduisent soit à des méthodes exactes et complètes, soit à des méthodes heuristiques. La complexité des premières rendait leur utilisation rédhibitoire pour des systèmes de dimension élevée (> 4). Les secondes souffraient des inconvénients liés à la non complétude. Récemment sont apparues de nouvelles méthodes de planification de type probabiliste qui permettent de faire face à la complexité exponentielle du problème. Ces méthodes vérifient une propriété de complétude probabiliste (garantissent en probabilité de trouver une solution en temps fini, lorsqu'elle existe) et s'avèrent très efficaces en pratique.

D'autre part, le problème de la planification de tâches de manipulation, est un problème combinatoirement plus complexe que celui de la planification du mouvement. En effet, il consiste non seulement à planifier les mouvements sans collision pour un robot, mais également le séquencement des différentes sous-tâches de prise et de pose d'objets. Un objet déplaçable est considéré comme un obstacle dans l'environnement. Cependant, sa position

*asahbani@laas.fr

peut être changée par un robot. Nos travaux portent sur le développement d'approches de type probabiliste pour la planification de tâches de manipulation.

2 Formulation géométrique du problème

La formulation géométrique retenue est celle introduite par R. Alami, J.P. Laumond, et T. Siméon dans [1]. Elle sera rappelée dans cette section puisque tout nos travaux sont basés sur cette formulation.

On se place pour la suite dans un environnement $3D$ contenant trois types d'objets: les obstacles statiques, le robot et les objets déplaçables. Pour le robot et pour chaque objet déplaçable, on associe un espace de configuration. L'espace de configuration pour chaque objet déplaçable est un espace de dimension 6, alors que celui du robot est de dimension n, où n est le nombre de degrés de liberté du robot. CS est le produit cartésien de tous ces espaces. Par ailleurs, CS_{free} représente l'ensemble des configurations dans lequel les différents objets ne sont pas en collision. Un chemin de manipulation est un chemin contraint dans CS_{free}. Il existe deux types de contraintes: de placements et de mouvements d'objets déplaçables.

2.1 Les contraintes de placements d'objets déplaçables

Les configurations de CS_{free} ne correspondent pas forcément à des configurations physiquement valides. En effet, un objet ne peut pas être considéré dans une configuration valide s'il flotte sans support dans l'air. Une configuration d'un objet déplaçable est considérée comme valide si elle est stable (au sens physique du terme). Notons par $PLACEMENT$ le sous ensemble de CS_{free} où tous les objets déplaçables sont dans des configurations stables. Ainsi, chaque objet déplaçable présent dans l'environnement a un ensemble de "pose" déterminé en fonction de la position des obstacles et des autres objets déplaçables. Notons que le sous-espace $PLACEMENT$ dépend fortement du contexte.

2.2 Les contraintes de mouvements d'objets déplaçables

Contrairement à un robot, un objet déplaçable ne peut pas bouger tout seul. Il ne peut être déplacé que par le robot. Pour déplacer un objet, le robot doit se mettre tout d'abord dans une position de "prise", puis activer l'opération d'attachement. Un attachement est une relation géométrique décrivant le placement de l'objet par rapport au robot. Notons par $GRASP$ le sous ensemble de CS_{free} dans lequel le robot se déplace de façon rigide avec l'objet déplaçable sans entrer en collision avec les obstacles ou avec les autres objets déplaçables. La relation géométrique qui définie une prise est considérée comme une donnée de notre problème. Cependant, le planificateur doit déterminer la configuration adéquate du robot vérifiant cette relation entre l'organe terminal et l'objet à déplacer.

2.3 Quelques définitions

- **Définition1**
  On appelle *chemin de transit* tout chemin de CS_{free} dans lequel le robot se déplace tout seul sans collision. Les objets déplaçables sont supposés dans des positions stables. On peut en déduire donc qu'un *chemin de transit* est inclus dans le sous-espace $PLACEMENT$.

- **Définition2**
  On appelle *chemin de transfert* pour l'objet déplaçable M_i tout chemin de CS_{free} dans lequel le robot déplace M_i sans entrer en collision ni avec les obstacles ni avec les autres objets déplaçables. Ces derniers sont supposés stationnaires dans des configurations stables.

- **Définition3**
  Un *chemin de manipulation* est un chemin dans CS_{free} composé d'une séquence alternée

de *chemins de transit* et de *chemins de transfert*. A chaque début (resp. fin) d'un *chemin de transfert*, le robot saisit (resp. pose) l'un des objet déplaçables.

2.4 Planifier des tâches de manipulation

En se basant sur les définitions introduites précédemment, on peut en déduire qu'un *chemin de transfert* et un *chemin de transit* sont connectés si et seulement si tous les deux ont des extrémités communes dans l'espace $PLACEMENT \cap GRASP$.
Le problème de planification de tâches de manipulation apparaît ainsi comme un problème de recherche de chemins dans les différentes composantes connexes du sous-espace $PLACEMENT \cap GRASP$ ou entre elles. L'exploration de l'espace $PLACEMENT \cap GRASP$ s'avère donc nécessaire pour la résolution d'un problème de planification de tâches de manipulation. On définit alors le *graphe de manipulation* comme étant le graphe dont les nœuds forment les composantes connexes de $PLACEMENT \cap GRASP$. Il existe deux types d'arêtes. Une arête de transfert (resp. de transit) liant deux nœuds indique qu'il existe un chemin de transfert (resp. de transit) connectant les deux configurations des composantes connexes associées.

3 État de l'art

Plusieurs algorithmes permettant la résolution de problème de planification de tâches de manipulation ont été proposés dans la littérature.
Alami, Laumond et Siméon présentent dans [1, 2] un schéma de résolution appliqué à deux instances de problèmes: cas de plusieurs objets manipulés par un polygone en translation et pour un bras à trois articulations rotoïdes sous l'hypothèse de nombre fini de prises et de poses, et cas d'un seul objet déplaçable dans un environnement polygonal avec un nombre fini de prises.
La solution de Koga et Latombe [3] consiste d'abord à planifier une trajectoire pour l'objet déplaçable, puis trouver une succession de *chemin de transfert* et de *chemin de transit* déplaçant l'objet le long de la trajectoire précalculée.
Ahuactzin et Mazer ont étendu l'algorithme de Fil d'Ariane afin de supporter des environnements dynamiques. L'extension est appelé "The Ariadne's clew algorithm" [4]. Il est basé sur des algorithmes génétiques. Il collecte des informations sur l'espace atteignable depuis la position initiale en plaçant des balises dans l'espace de recherche. En effet les balises représentent une configuration de prise stable atteignable à partir de la configuration initiale. Pour chaque nouvelle balise placée, le planificateur essaie d'atteindre la configuration but en exécutant un chemin local [5]. L'idée principale derrière cette algorithme est d'eviter de construire l'espace des configurations en conduisant les recherches dans l'espace des trajectoires.
Une autre méthode proposée par Nielsen et Kavraki est une extension des méthodes probabilistes (PRM: probabilistic roadmap approach). Cette méthode est appelée "Fuzzy PRM" [6]. Elle consiste à construire un graphe dont les arêtes sont annotées par une probabilité. Cette probabilité est une estimation de chance que l'arête est actuellement faisable (sans collision).
Après cette brève présentation des travaux similaires, la section 4 ainsi que la 5, seront consacrées à la présentation des algorithmes que nous avons développé.

4 Résoudre un problème simple de manipulation

Considérons le scénario illustré dans la Figure 1. Le robot partant de sa position initiale dans la première pièce, doit rejoindre sa position finale dans l'autre pièce. L'unique accès à la deuxième pièce est bloqué par une chaise.
Ce type de tâche peut être décomposé en trois sous-tâches: "*Va à la chaise*", "*Prend la chaise et pose la de façon à accéder à l'autre pièce*", "*Va à ta configuration but*". Chaque sous-tâche est traitée selon une approche probabiliste. Ainsi, trois graphes sont calculés tenant compte de

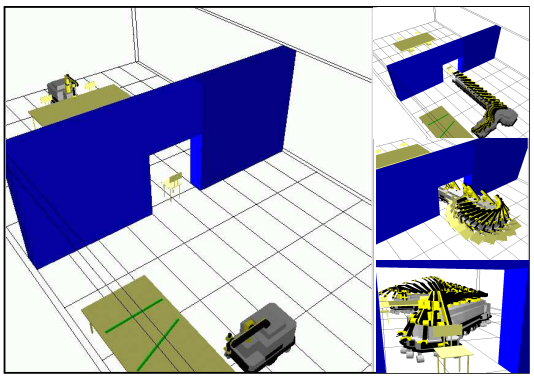

FIG. 1 – *Exemple simple de prise et de pose.*

l'état courant de l'environnement. Le premier graphe (Figure 2(a)) représente le recouvrement de l'espace *GRASP*. Le graphe est constitué d'une seule composante connexe, ce qui implique que le robot peut poser la chaise là ou il veut, tout en respectant les contraintes de placements. Le deuxième graphe est calculé en considérant que la chaise occupe la configuration indiquée dans la Figure 2(b). Ce graphe recouvre donc, le sous-ensemble de *PLACEMENT* correspondant au placement initial de la chaise. Le troisième graphe présenté dans la Figure 2(c) couvre de même le sous-ensemble de *PLACEMENT* correspondant au placement final de la chaise. Le fait que dans le deuxième ainsi que dans le troisième cas, les graphes correspondant sont constitués de deux composantes connexes montrent que la chaise bloque bien le passage entre les deux pièces.

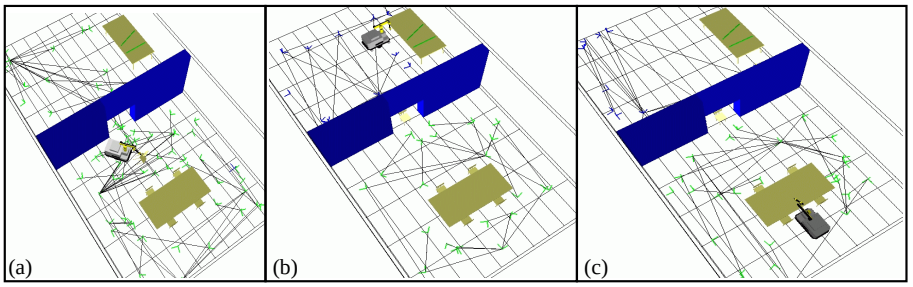

FIG. 2 – *Les trois graphes calculés.*

Dans ce type de problème, l'opérateur doit définir la position initiale et finale du robot ainsi que les différentes prises possibles des objets déplaçables. L'algorithme développé calcule les trois graphes décrits précédemment puis génére une trajectoire valide. Il est à noter qu'il faut rajouter aux graphes calculés des nœuds permettant la connexion et le passage d'un graphe à un autre. En outre, ces nœuds permettent la connexion entre *un chemin de transfert* et *un chemin de transit*.

5 Une nouvelle approche basée sur la notion de graphe dynamique

La génération d'un graphe est coûteuse en terme de temps de calcul. L'idée principale de cette approche est de minimiser le nombre de graphes calculés. Elle combine deux méthodes: réseaux probabiliste basés sur la visibilité et RRT (Rapidly-exploring Random Trees).

5.1 La méthode des réseaux probabilistes basés sur la visibilité

La méthode des réseaux probabilistes est une méthode de planification de mouvement pour des environnements statiques, complète en probabilité et qui n'est pas restreinte à une classe particulière de robots. Elle procède en deux phases: une phase d'apprentissage et une phase de recherche [7]. Lors de la phase d'apprentissage, un graphe est construit selon un tirage aléatoire dans l'espace des configurations. Les nœuds de ce graphe représentent des configurations libres (sans collision) du robot dans l'environnement. Les arêtes représentent des chemins faisables par le robot entre les configurations. Ces chemins sont construits selon la méthode locale adoptée. Après la construction du graphe, la phase de recherche est entamée. Etant données une configuration de départ et une configuration d'arrivée, on les relie au graphe par des chemins locaux faisables. Une fois liées, il nous reste qu'à chercher un chemin dans le graphe connectant les deux nœuds ajoutés. La méthode des réseaux probabilistes basée sur la visibilité [8] est une variante des méthodes probabilistes: un nœud n'est ajouté au graphe que s'il n'appartient à aucun domaine de visibilité des nœuds déjà présent ou s'il permet de relier plusieurs composantes connexes du graphe.
Le premier avantage de l'approche est de définir un critère d'arrêt explicite de l'algorithme de construction du graphe, critère absent dans les approches jusqu'ici développées. Par ailleurs la taille réduite du graphe conduit à un gain significatif en temps de calcul.

5.2 RRT: méthode d'exploration rapide

Le formalisme d'exploration rapide et aléatoire des arbres a été introduit par LaValle dans [9, 10]. Cette méthode consiste a faire évoluer deux arbres ayant comme racine les deux configurations qu'on cherche à connecter. La construction des arbres se fait de façon incrémentale dans le but de réduire la distance d'un point tiré aléatoirement aux deux arbres. Les deux arbres progressent l'un vers l'autre selon une heuristique simple. Cette méthode est adaptée au cas d'une recherche unique d'un chemin dans un environnement donné. Par ailleurs, cette méthode est complète en probabilité.

5.3 Notre approche

Au début, on calcule un graphe couvrant CS_{free} sans tenir compte de la présence d'objets déplaçables dans l'environnement. Afin de valider un *chemin de transit*, pour une répartition donnée des objets déplaçables dans l'environnement, le graphe est automatiquement mis à jour. Chaque nœud puis chaque arête sont testés afin de détecter les éventuelles collisions avec les positions courantes des objets déplaçables. Si un nœud est détecté en collision avec l'un des objets déplaçables, toutes les arêtes sortantes ou entrantes à ce nœud sont supprimées ainsi que le nœud correspondant. Cependant, si une arête est détectée en collision, sans que les nœuds (extrémités) ne le soient, Cette arête est marquée. La recherche de chemin se poursuit en utilisant le graphe ainsi marqué. Trois cas de figures peuvent se présenter. Le premier cas est de trouver un chemin ne comprenant aucune arête marquée, le chemin est donc validé et le problème est résolu. Le deuxième cas est de ne pas trouver un chemin, cela implique que le problème ne possède pas de solution. Finalement le dernier cas est lorsque le chemin contient au moins une arête marquée. Dans cette situation, l'algorithme essaie de résoudre le problème localement. La méthode RRT, décrite ci dessus, sert à remplacer le fragment en collision dans l'arête par un chemin tenant compte de la position actuelle des différents objets déplaçables. La complétude en probabilité de notre approche découle directement du fait que les deux méthodes utilisées, à savoir visibilité PRM et RRT, sont complètes en probabilité. La même approche est utilisée pour la recherche d'un *chemin de transfert*. Le graphe initialement calculé est mis à jour en considérant qu'il y a un objet déplaçable attaché au robot et en tenant compte de la position actuelle des autres objets déplaçables (cf. Figure 3).

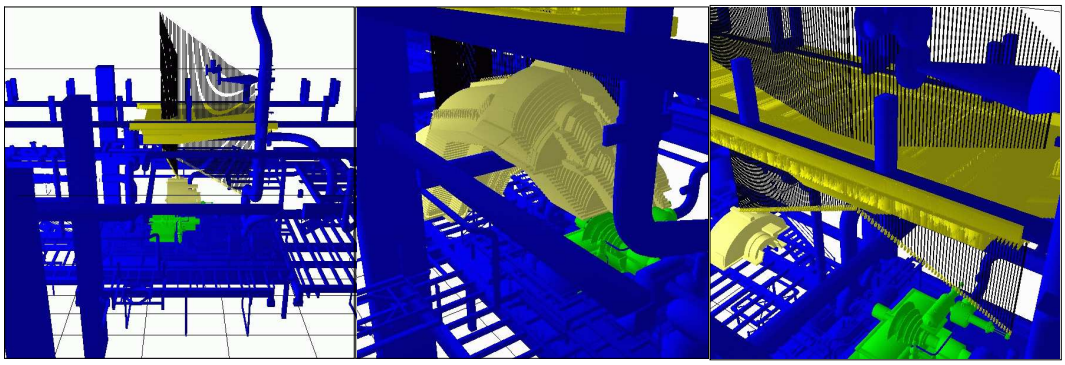

FIG. 3 – *Transport d'un couvercle d'une turbine nucléaire (modèle EDF).*

6 Conclusions et Perspectives

Nous avons présenté dans cet article deux types d'approches pour résoudre des problème de planification de tâches de manipulation. Il est à noter que les deux approches gardent la propriété de complétude en probabilité héritée des méthodes combinées.
Une autre approche est en cours de développement. Basée sur l'exploration de l'espace $GRASP \cap PLACEMENT$, elle spécifie directement les contraintes de déplacements au niveau de la définition des objets déplaçables. L'intérêt de cette nouvelle approche est d'étendre la notion d'ensemble discret de placement d'objets déplaçables adoptée par les approches présentées à un ensemble continu.

Références

[1] R. Alami, J.P. Laumond, and T. Siméon., "Two manipulation planning algorithms," *The first Workshop on the Algorithmic Foundations of robotics.(WAFR'94),San Fransisco*, pp. 109–125, February 1994.

[2] R. Alami, T. Siméon, and J-P.Laumond., "A geometrical approach to planning manipulation tasks. the case of discrete placements and grasps.," *In Robotics Research: The Fifth International Symposium,Tokyo*, August 1989.

[3] Y. Koga and J.-C. Latombe, "On multi-arm manipulation planning.," *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 945–952, 1994.

[4] P. Bessiere, J.M. Ahuactzin, E.-G. Talbi, and E. Mazer., "The ariadne's clew algorithm: Global planning with local methods.," *Proc. IEEE Int. Conf. Intel. Rob. Syst. (IROS)*, vol. 2, pp. 1373–1380, 1993.

[5] J.-M. Ahuactzin, K. Gupta, and E. Mazer., "Manipulation task planning for redundant robots: A practical approach.," *International Journal of Robotics Research, MIT Press*, pp. 731–741, 1998.

[6] C.L. Nielsen and L.E. Kavraki., "A two level fuzzy prm for manipulation planning.," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2000.

[7] L. Kavraki, P. Svestka, J.C. Latombe, and M. Overmars., "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Int'l Conf. on Robotics and Automation (ICRA'1996)*, pp. 566–580, 1996.

[8] C. Nissoux, T. Simeon, and J.-P. Laumond., "Visibility based probabilistic roadmaps," *IEEE International Conference on Intelligent Robots and Systems*, pp. 1316–1321, 1999.

[9] Steven M. LaValle., "Rapidly-exploring random trees: A new tool for path planning.," *Technical Report No. 98-11, Dept. of Computer Science, Iowa State University*, 1998.

[10] J.J. Kuffner and S.M. LaValle., "Rrt-connect: An efficient approach to single-query path planning.," *IEEE Int'l Conf. on Robotics and Automation (ICRA'2000),San Francisco*, 2000.