Analyse de l'article « On the Relationship Between Classical Grid Search and Probabilistic Roadmaps »

Quang-Cuong Pham

31 janvier 2005

1 Le problème de planification de mouvement

1.1 Présentation

Soit un robot à d degrés de liberté. Le problème de planification de mouvement se modélise ainsi :

- Espace des configurations $\mathcal{C}$ de dimension d. Espace libre $\mathcal{C}_{free}$.
- Une requête est une paire de configurations $(q_{init}, q_{goal}) \in \mathcal{C}^2$.
- Question : trouver un chemin continu $\tau : [0, 1] \to \mathcal{C}_{free}$ tel que $\tau(0) = q_{init}$ et $\tau(1) = q_{goal}$.

On suppose qu'on ne dispose pas de représentation explicite de $\mathcal{C}$, mais d'un *détecteur de collision* qui peut tester si une configuration q appartient à $\mathcal{C}_{free}$.

1.2 Méthodes basée sur l'échantillonnage

Il existe des méthodes combinatoires pour planifier le mouvement (voir le cours). On étudie ici les méthodes basées sur l'échantillonnage. Voici l'algorithme générique pour la phase de « preprocessing ».

On construit un graphe G dont les sommets représentent des points de $\mathcal{C}_{free}$ (l. 3, l. 4). On essaie de voir si on peut aller de ce point vers des points *voisins* à l'aide d'un planneur local (l. 6). Si oui, on ajoute l'arête (q, v) au graphe G (l. 7).

Pour traiter une requête, on considère q_{init} et q_{goal} comme deux nouveaux sommets qu'on essaie de connecter à leurs voisins comme dans les lignes 5, 6, 7. Ensuite, il suffit de chercher s'il existe un chemin joignant ces deux nouveaux sommets dans le graphe G.

Plusieurs remarques :

NEW-POINT() : L'échantillonnage consiste précisément en cette fonction. Dans une méthode probabiliste, on tire un nouveau point *au hasard* dans $\mathcal{C}_{free}$. Mais on peut aussi choisir ce nouveau point dans

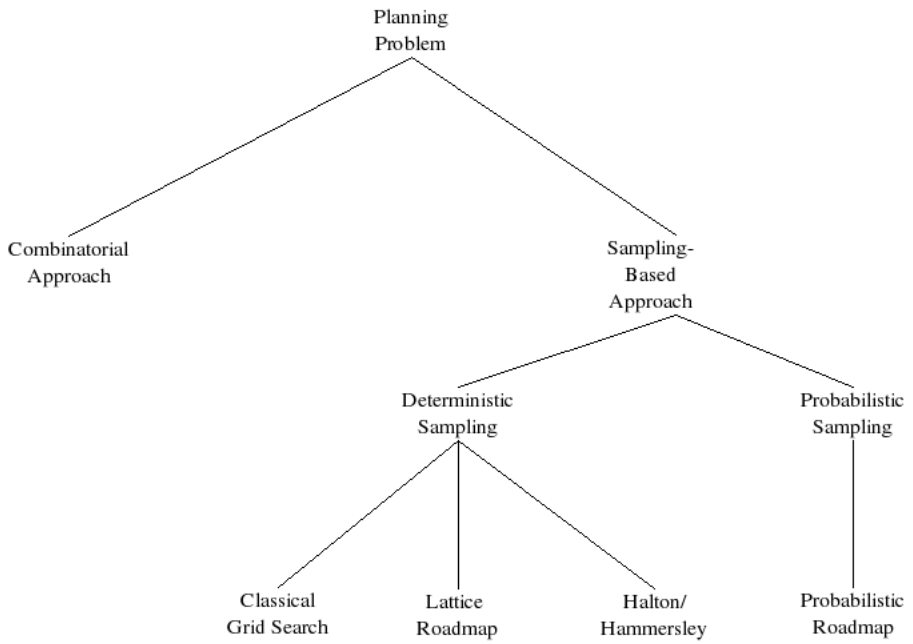

FIG. 1 – Les méthodes de planification de mouvement

une séquence de Halton, de Hammersley, ou dans un réseau, etc. (voir section 2).

NEIGHBORHOOD(G, q) : Cette fonction retourne un ensemble de points « voisins » de q dans le graphe. Dans les méthodes non structurées (probabiliste, Halton, Hammersley) on peut chercher tous les points dans un rayon R de q, ou alors les K plus proches voisins de q. Dans les méthodes structurées (grille, réseau), on peut mettre à profit la structure pour se passer de cette recherche, qui peut coûter assez cher (par exemple dans une grille de dimension 2, les voisins d'un sommet q peuvent être les quatre points « en haut », « en bas », « à gauche », « à droite » de q).

CONNECT(q, v) : Cette fonction teste si on peut aller « localement » de q à v. En pratique, on peut examiner si les points qui sont sur la ligne droite joignant q et v appartiennent ou non à $\mathcal{C}_{free}$.

Version paresseuse : Dans la version décrite ci-dessus, on a distingué deux phases : une phase (complexe) de « preprocessing » où on a construit un graphe approximant la « structure combinatoire » de $\mathcal{C}_{free}$, et une phase (simple) de réponse aux requêtes, qui consiste essentiellement en une recherche de chemin dans un graphe. Cette construction convient donc particulièrement pour répondre à un grand nombre

Algorithm 1 Build Roadmap

```
INITIALIZE(G)
for i = 1 to N do
    q ← NEW-POINT()
    ADD-VERTEX(G, q)
    for all v ∈ NEIGHBORHOOD(G, q) do
        if CONNECT(q, v) then
            ADD-EDGE(G, (q, v)))
        end if
    end for
end for
```

de requêtes dans un environnement fixé. Or si l'on veut répondre seulement à un nombre réduit de requêtes, on n'a pas besoin de construire tout le graphe. En pratique, on ajoute au fur et à mesure des sommets et des arêtes dans le graphe tant que q_{init} et q_{goal} sont dans des composantes connexes distinctes.

Suite vs ensemble : Dans le PRM et les QRM, on a des suites de points, alors que dans le SGS et le LRM on a des ensembles de points (voir discussion plus loin).

2 Comparaison des planifieurs par échantillonnage

2.1 Critères de comparaison

Intuitivement, la qualité d'un échantillonnage se voit à la distribution plus ou moins uniforme des points : si beaucoup de points sont concentrés en un endroit, ou si au contraire il existe des grandes régions vierges de points, l'échantillonnage est mauvais. Soit $X = [0,1]^d$ l'espace qu'on veut échantillonner, et P un ensemble d'échantillons (points de X). Pour mesurer cette notion d'uniformité, il y a deux quantités :

Disparité : Soit $\mathcal{R}$ une collection de sous-ensembles de X (souvent $\mathcal{R}$ est l'ensemble des pavés de X). Pour $R \in \mathcal{R}$ on note $\mu(R)$ la mesure de Lebesgue de R. La disparité de P par rapport à $\mathcal{R}$ est :

$$D(P, \mathcal{R}) = \sup_{R \in \mathcal{R}} \left| \frac{\sharp(P \cap R)}{\sharp P} - \mu(R) \right|$$

Intuitivement, si par exemple $\mu(R) = 1/5$, on s'attend à ce que R contiennent à peu près $1/5$ des points de P.

Dispersion : Soit d une distance sur X. La dispersion de P par rapport à d est :

$$\delta(P,d) = \sup_{x\in X} \min_{p\in P} d(x,p)$$

La dispersion peut être interprétée comme le rayon de la plus grande boule possible ne contenant aucun point de P.

Les deux notions de disparité et de dispersion proviennent de l'intégration numérique. Pour les problèmes de planification, nous verrons que c'est la dispersion qui sera la notion la plus appropriée.

On constate aussi que $\delta(P,l^\infty)^d \leq D(P,\text{pavés})$.

2.2 Dispersion et complétude des planifieurs

Pour $q \in \mathcal{C}_{free}$, notons l'ensemble des points visibles à partir de q : $V(q) = \{q' \in \mathcal{C}_{free} : \forall\lambda \in [0,1], \lambda q + (1-\lambda)q' \in \mathcal{C}_{free}\}$.

Notons $\Gamma(\mathcal{C}_{free})$ l'ensemble des requêtes, et $\Gamma_s(\mathcal{C}_{free})$ l'ensemble des requêtes pour lesquelles il existe une solution.

Soit $\gamma \in \Gamma_s(\mathcal{C}_{free})$. Pour tout chemin $\tau : [0,1] \to \mathcal{C}_{free}$ satisfaisant γ et tout $r > 0$, on peut associer un *tube* $B = \bigcup_{s\in[0,1]} B(\tau(s), r)$. La *largeur* du tube est $2r$.

Parmi tous ces tubes, notons $B(\gamma)$ le tube de largeur maximale tel que $B(0) \subset V(q_{init})$ et $B(1) \subset V(q_{goal})$.

Notons $w(\mathcal{C}_{free}) = \inf_{\gamma\in\Gamma_s(\mathcal{C}_{free})} w(\gamma)$.

Pour $x > 0$, notons $\Psi(x) = \{\mathcal{C}_{free} : w(\mathcal{C}_{free}) \geq x\}$. On dit qu'un algorithme est complet pour $\Psi(x)$ s'il répond correctement à toute requête de tout $\mathcal{C}_{free} \in \Psi(x)$.

Proposition 1 *Soit un algorithme dont l'échantillonnage a une dispersion δ. Alors l'algorithme est complet pour $\Psi(4\delta)$.*

Proposition 2 *Soit un algorithme dont l'échantillonnage a une dispersion δ. Alors il n'est pas complet pour $\Psi - \Psi(\delta)$.*

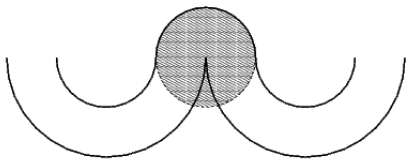

FIG. 2 – Un exemple pour montrer la Proposition 2

2.3 Les différentes méthodes d'échantillonnage

Une manière de répartir 8 points uniformément dans $[0,1]$ est de les mettre aux abscisses $0.000, 0.001, 0.010, 0.011, \ldots$

Séquence de *Van der Corput* : on inverse l'ordre des bits pour augmenter la distance entre deux points consécutifs : 0.000, 0.100, 0.010, 0.110, 0.001, 0,101, 0.011, 0.111.

On peut aussi le faire en base $p \geq 2$:

– On écrit i en base p : $i = a_0 + pa_1 + p^2 a_2 + ...$

– On calcule $r_p(i) = \frac{a_0}{p} + \frac{a_1}{p^2} + \frac{a_2}{p^3} + \cdots \in [0,1]$

On construit ainsi des échantillonnages quasi-aléatoires :

Points de Halton en dimension d : On choisit d nombres premiers entre eux $p_1, \ldots, p_d$ (généralement les d premiers nombres premiers). Le $i^{\text{ème}}$ élément est : $(r_{p_1}(i), \ldots, r_{p_d}(i))$.

Points de Hammersley en dimension d : On choisit $d-1$ nombres premiers entre eux $p_1, \ldots, p_{d-1}$. Le $i^{\text{ème}}$ élément d'un ensemble de N points est : $(\frac{i}{N}, r_{p_1}(i), \ldots, r_{p_{d-1}}(i))$.

Les échantillonnages structurés :

Grille de Sukharev : Soit $k = N^{1/d}$. On divise le cube $[0,1]^d$ en $k \times k \times \cdots \times k$ petits cubes au centre desquels on place un point.

Réseaux : C'est une généralisation des grilles avec des axes non orthonormés (utile quand on sait que l'environnement contient beaucoup d'alignements ...).

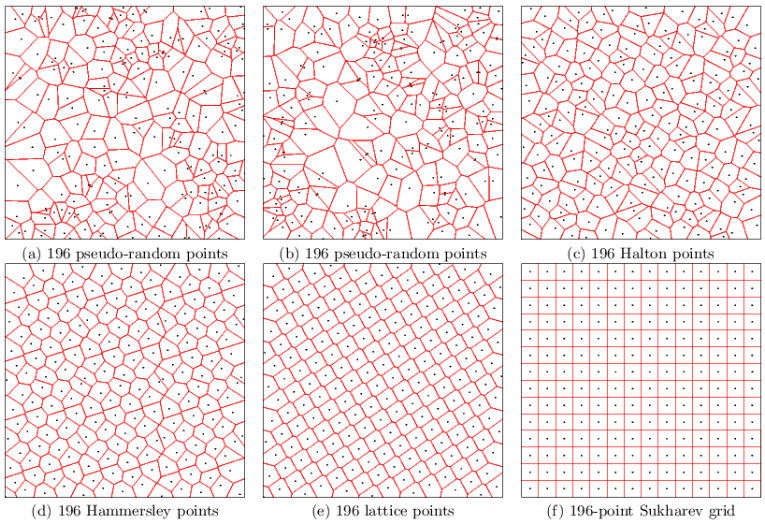

FIG. 3 – Les méthodes d'échantillonnage

2.4 Comparaison théorique

On parle ici de la dispersion l^∞, au lieu de la dispersion euclidienne, plus difficile à estimer. Cependant, comme on est en dimension finie il y a équivalence entre ces deux notions.

Sukharev a montré que pour tout P : $\delta(P) \geq \frac{1}{2\lfloor(\sharp P)^{1/d}\rfloor}$.

En ce qui concerne les méthodes qu'on a rencontrées :

- Pour N puissance exace de d, la grille de Sukharev atteint cette borne.
- Généralement grille et réseaux sont proches de cette borne
- Halton et Hammersley sont en $b(d)N^{-1/d}$ où $b(d)$ ne dépend pas de N mais croît très vite avec d.
- L'échantillonnage probabiliste est en $O((\log N)^{1/d}N^{-1/d})$.

2.5 Résultats expérimentaux

Les résultats expériemetaux confirment qualitativement l'analyse théorique.

Les grilles ont le désavantage que les choix du nombre de points par axe est limité.