Feuille d'exercices #5 — Planification

IFT 3345

Question 1. ***Champs de potentiels artificiels (potential fields)***

Un robot se déplace dans $\mathbb{R}^2$. Sa position actuelle est

$$x = (1, 1)$$

L'objectif est situé en

$$x_{\text{goal}} = (3, 1)$$

et un obstacle est situé en

$$x_{\text{obs}} = (1, 3).$$

On définit les potentiels suivants :

— Potentiel attractif :

$$U_{\text{att}}(x) = \frac{1}{2}\|x - x_{\text{goal}}\|^2$$

— Potentiel répulsif (pour chaque obstacle) :

$$U_{\text{rep}}(x) = \begin{cases} \frac{1}{2}\left(\frac{1}{d(x)} - \frac{1}{d_0}\right)^2 & \text{si } d(x) \leq d_0 \\ 0 & \text{sinon} \end{cases}$$

où $d(x) = \|x - x_{\text{obs}}\|$ et $d_0 = 3$.

(a) Calculer la force attractive :

$$F_{\text{att}} = -\nabla U_{\text{att}}(x).$$

(b) Calculer la force répulsive :

$$F_{\text{rep}} = -\nabla U_{\text{rep}}(x).$$

(c) Calculer la force totale :

$$F = F_{\text{att}} + F_{\text{rep}}.$$

(d) Dans quelle direction générale le robot va-t-il se déplacer ?

Question 2. ***Rapidly-Exploring Random Trees (RRT)***

On considère un robot se déplaçant dans un environnement 2D. La position initiale est :

$$x_{\text{start}} = (0, 0)$$

Un obstacle rectangulaire occupe la région :

$$2 \leq x \leq 4, \quad 1 \leq y \leq 3$$

La taille de pas est fixée à $\delta = 1$ (Avancer depuis ce nœud en direction du point échantillonné d'une distance $\delta = 1$).

On vous donne la séquence de points échantillonnés suivante :

$$x_1 = (3, 0), \quad x_2 = (3, 3), \quad x_3 = (5, 2), \quad x_4 = (6, 0)$$

1. À partir de x_{start}, construire l'arbre RRT étape par étape.
2. Pour chaque point x_i, indiquer :
   — le nœud le plus proche,
   — le nouveau nœud ajouté (si applicable),
   — si l'arête est rejetée à cause d'une collision.
3. Dessiner l'arbre final.

Question 3. ***Planification avec Probabilistic Roadmaps (PRM)***

On considère un robot se déplaçant dans un environnement 2D avec des obstacles. Les positions de départ et d'arrivée sont x_{start} et x_{goal}.

Écrire un algorithme en pseudo-code

```
function PRM(x_start, x_goal, N, d)
```

pour calculer un chemin de x_{start} à x_{goal} en utilisant la méthode PRM.

Votre algorithme doit :

— Échantillonner des configurations dans l'espace libre
— Construire un graphe de roadmap $G = (V, E)$ (vous pouvez ajouter des sommets et des arêtes au graphe avec les fonctions `G.add_node(v)` et `G.add_edge(v1, v2)`).
— Connecter chaque nœud à ses voisins dans un rayon d
— Ajouter x_{start} et x_{goal} au graphe
— Calculer le chemin final

On suppose que les fonctions suivantes sont disponibles :

— `SampleFree()` : retourne une configuration aléatoire dans l'espace libre
— `CollisionFree(x1, x2)` : retourne `true` si le segment entre x_1 et x_2 est sans collision
— `Near(x, G.V, d)` : retourne l'ensemble des nœuds de V à une distance inférieure à d de x
— `ShortestPath(G, start, goal)` : retourne le plus court chemin dans le graphe G

Question 4.

Wavefront Planner

On considère un robot évoluant dans une grille 2D discrète. Chaque cellule est soit libre, soit occupée par un obstacle.

Le **wavefront planner** fonctionne comme suit :

— On initialise la cellule objectif avec la valeur 0.
— On propage ensuite des valeurs entières dans la grille :
   — chaque cellule libre voisine reçoit une valeur égale à la valeur de la cellule courante + 1
— On répète jusqu'à ce que toutes les cellules accessibles aient été visitées.
— Pour planifier un chemin depuis x_{start}, on suit à chaque étape le voisin ayant la plus petite valeur.

(a) Ce planificateur est-il **complet** ? Justifier.
(b) Ce planificateur est-il **optimal** ? Justifier.

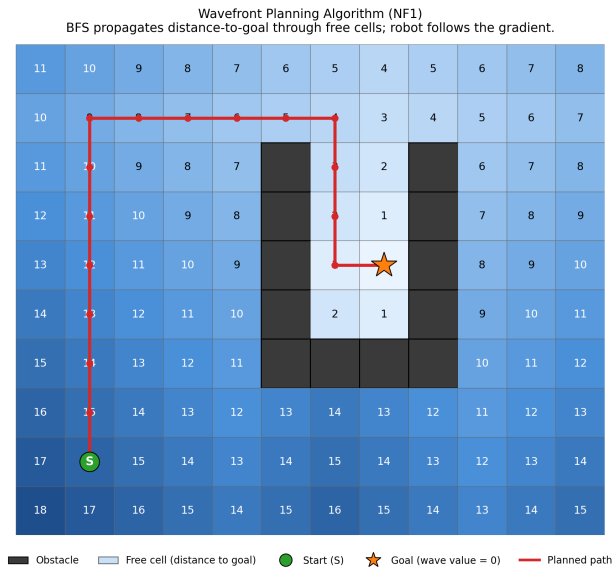

FIGURE 1 – Le **wavefront planner**

(c) Ce planificateur est-il **réactif** ou **délibératif** ? Justifier.

(d) Formuler le problème sous forme d'un **processus de décision markovien (MDP)** :
- — définir l'ensemble des états
- — définir l'ensemble des actions
- — définir la fonction de transition
- — définir la fonction de coût (ou récompense)