

Planification de trajectoires pour la tomographie robotisée

Hugo Rémin
Nicolas Delanoue, Sébastien Lagrange



Le jeudi 8 décembre 2022
Séminaire du LARIS

Table des matières

- 1 Contexte de Travail
- 2 Introduction : espace de travail et espace de configurations
- 3 Etude de C_{free} et planification de trajectoires entre deux points
 - Méthodes déterministes
 - Méthodes probabilistes
- 4 Planification de trajectoires entre n points
 - Travelling Salesman Problem (TSP)
 - Connexité (Algorithme de Tarjan)
- 5 Conclusion et perspectives
 - Multiplicité des solutions
 - Connexité généralisée

Contexte

- Partenariat entre le CEA TECH Nantes et le LARIS
- Support du CEA LIST pour la tomographie



Saclay



Angers



Nantes

Tomographie

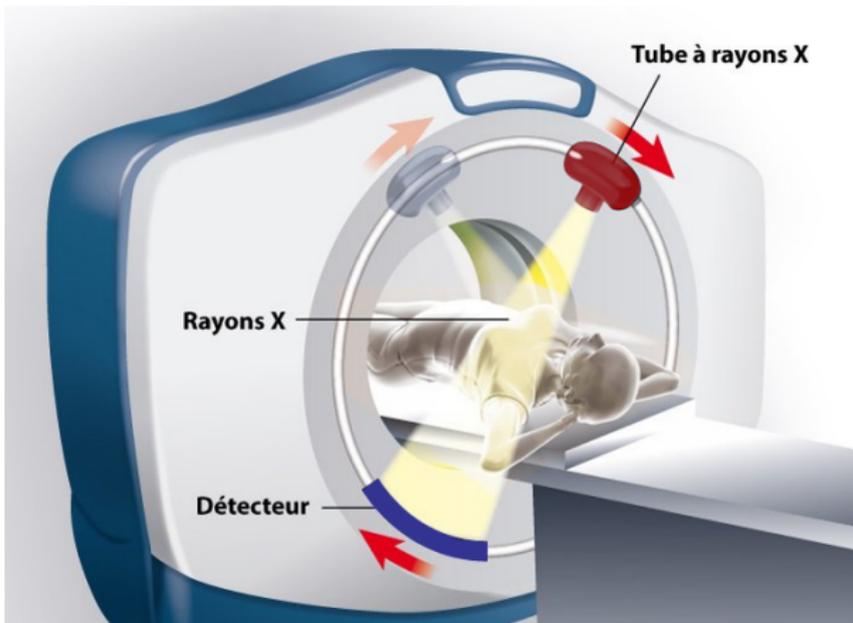


FIGURE – Fonctionnement de la tomographie classique

Cellule robotisée



FIGURE – Cellule robotisée

Description :

- Deux robots 6 axes
- Deux glissières (5 mètres)
- Récepteur et émetteur de rayons X

Cellule robotisée

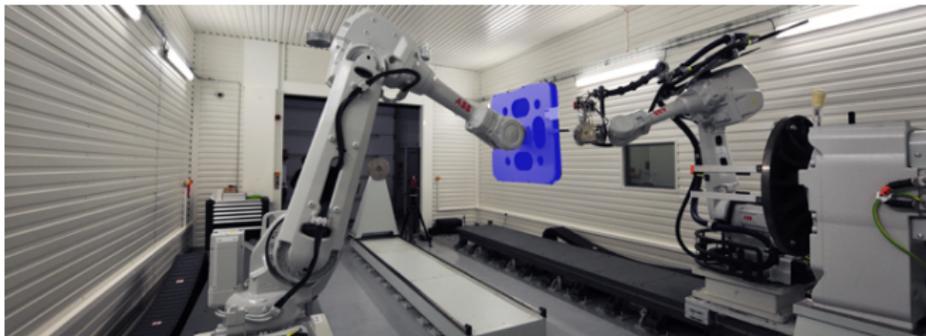


FIGURE – Cellule robotisée

Description :

- Deux robots 6 axes
- Deux glissières (5 mètres)
- Récepteur et émetteur de rayons X

Cellule robotisée

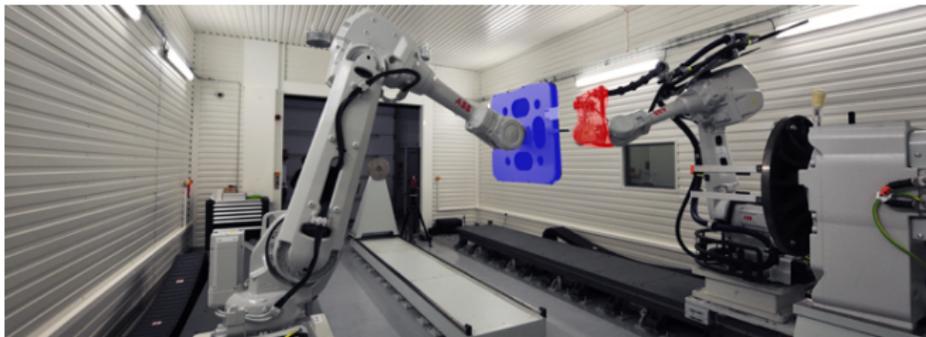


FIGURE – Cellule robotisée

Description :

- Deux robots 6 axes
- Deux glissières (5 mètres)
- Récepteur et émetteur de rayons X

Cahier des charges

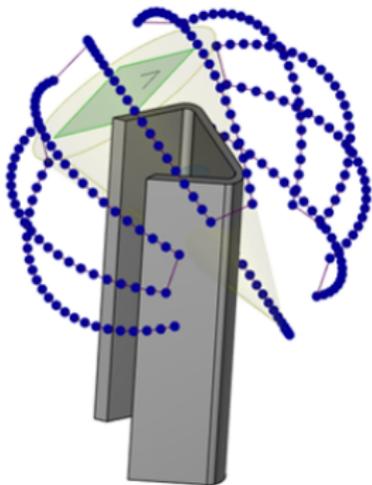
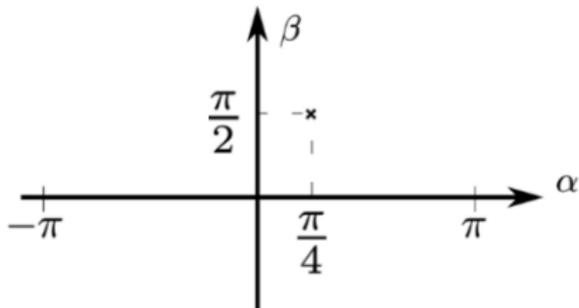


FIGURE – Exemple de position à atteindre pour réaliser des clichés

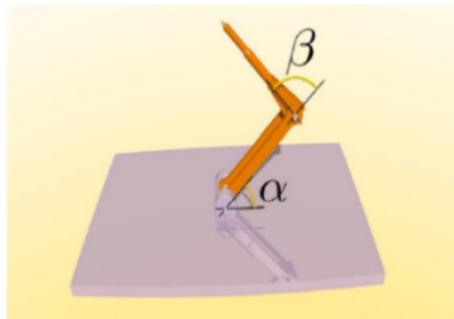
Objectifs

- Automatisation de la plateforme d'imagerie RX robotisée :
 - Etape 1 : Trouver les n points à visiter (CEA List)
 - Etape 2 : Calculer la trajectoire des robots permettant l'acquisition des n clichés

Modélisation robotique



Espace des configurations \mathbf{C}



Espace de travail \mathbf{W}

Modèle Géométrique

Direct :

$$f: \mathbf{C} \rightarrow \mathbf{W}$$

$$f(q) \mapsto x$$

Inverse :

$$f^{-1}: \mathbf{W} \rightarrow 2^{\mathbf{C}}$$

$$f^{-1}(x) \mapsto \{q\}$$

Formalisation d'une trajectoire

Définition

On appelle trajectoire une fonction γ continue allant d'un point de départ P_1 à un point d'arrivée P_2 .

$$\begin{aligned} \gamma: [0, 1] &\rightarrow C \\ t &\mapsto q \end{aligned}$$

$$\text{telle que } \begin{cases} \gamma(0) = P_1, \\ \gamma(1) = P_2, \\ \forall t \in [0, 1], \quad \gamma(t) \in C_{free}. \end{cases}$$



Caractérisation de C_{free}

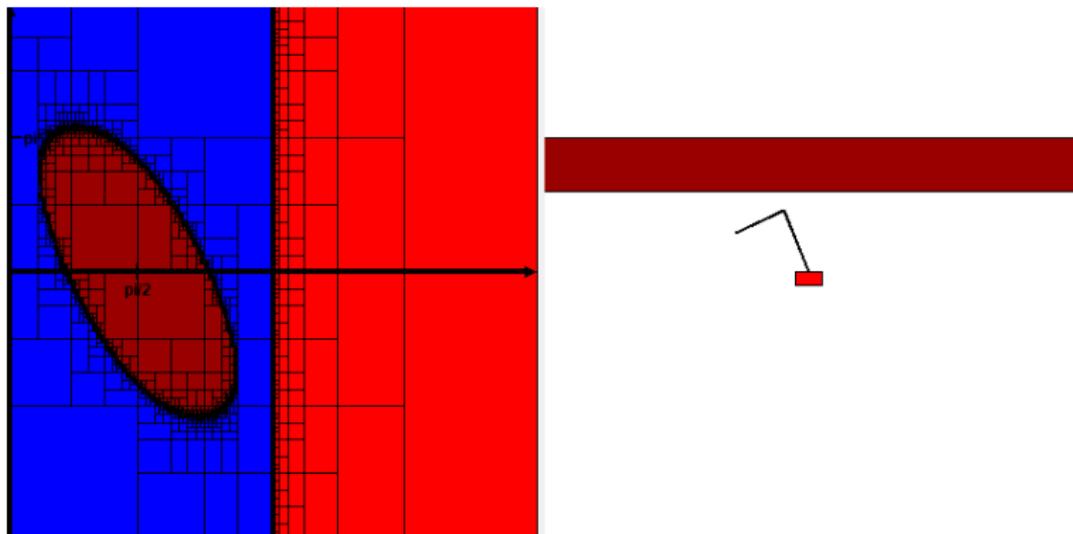


FIGURE – Exemple avec un robot à deux articulations rotatoires

Etude de C_{free}

Deux types de méthodes

Méthodes déterministes :

- Caractérisent complètement C_{free} ,
- À base de décomposition cellulaire,
- Complexité exponentielle avec le nombre de dimension.

Etude de C_{free}

Deux types de méthodes

Méthodes déterministes :

- Caractérisent complètement C_{free} ,
- À base de decomposition cellulaire,
- Complexité exponentielle avec le nombre de dimension.

Méthodes probabilistes :

- Approximent C_{free} ,
- À base d'échantillonnage de C_{free} ,
- Complexité inchangée avec le nombre de dimension, seul le nombre de points augmente.

Méthodes déterministes

Propriétés des méthodes déterministes

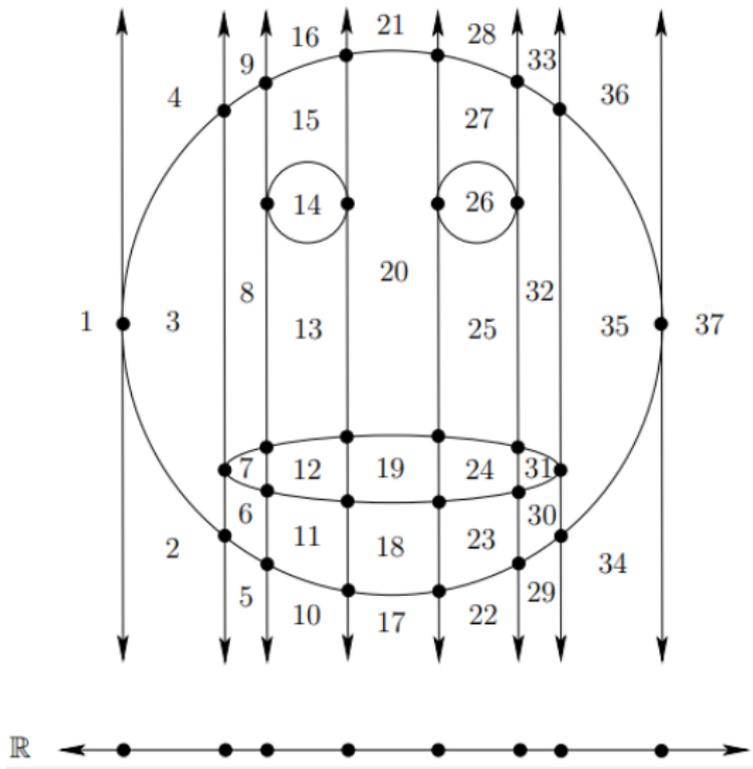
- Caractérisent complètement C_{free} ,
- Trouvent la trajectoire optimale si elle existe,
- Lente.

Cylindrical Algebraic Decomposition (CAD)

Principe

- 1 Développer une séquence de projections diminuant la dimension n de l'ensemble semi-algébrique d'un.
- 2 La projection atteint \mathbb{R} , et un polynôme univarié est obtenu pour lequel les zéros sont aux endroits critiques où les frontières des cellules doivent être formées.
- 3 La séquence est ensuite inversée, et les décompositions sont formées de $\mathbb{R}, \mathbb{R}^2, \dots, \mathbb{R}^n$

Cylindrical Algebraic Decomposition (CAD)



Set inversion via interval analysis (SIVIA)

Principe

- 1 Soit le pavé courant P ,
- 2 Si P est démontré inclus dans C_{free} alors il est bleu,
- 3 Si p est démontré non-inclus dans C_{free} alors il est rouge,
- 4 Sinon on scinde P en deux et on recommence récursivement.

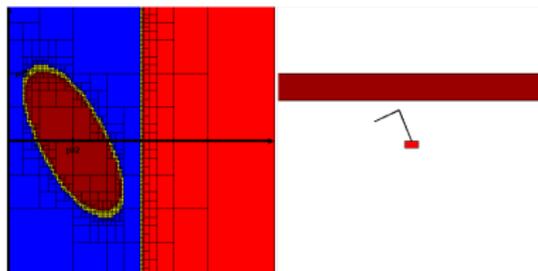


FIGURE – Exemple en deux dimensions

Méthodes probabilistes

Propriétés des méthodes probabilistes

- Complètes en probabilité i.e. $\lim_{i \rightarrow \infty} P(\gamma | \exists \gamma) = 1$,
- Ne trouvent pas forcément la même solution à chaque exécution,
- Rapide.

Caractéristiques de RRT et PRM

- RRT est dit "single query" i.e. on crée un arbre pour chaque trajectoire voulue
- PRM est dit "multiple query" i.e. on répond à plusieurs requêtes avec la même carte
- PRM est plus lent lorsqu'il y a peu de requête
- RRT a une variante convergeant asymptotiquement à la trajectoire optimale

Rapidly-exploring random tree (RRT)

Obstacles
Modèle géo- → **RRT** → Arbre
métrique

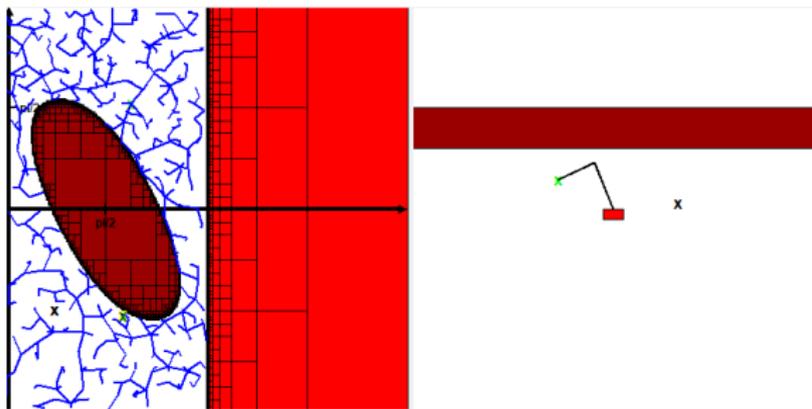


FIGURE – Algorithme du RRT - $n = 500$

Rapidly-exploring random tree (RRT)

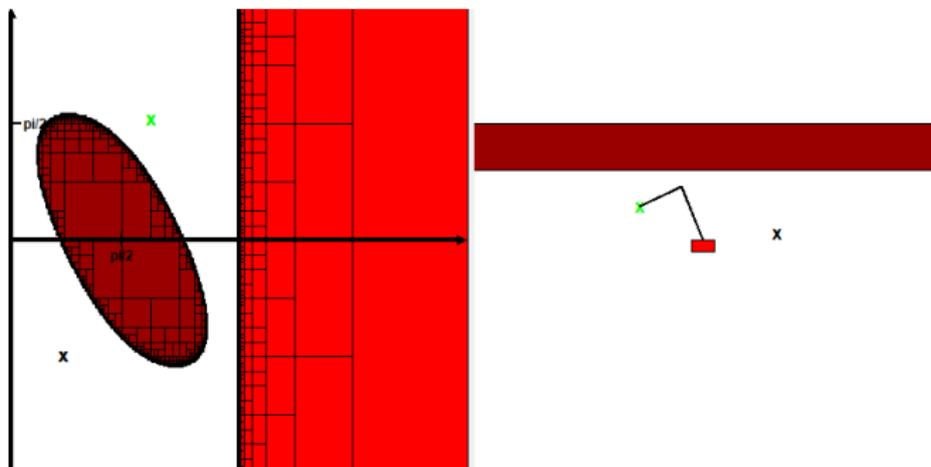


FIGURE – Algorithme du RRT - $n = 1$

Rapidly-exploring random tree (RRT)

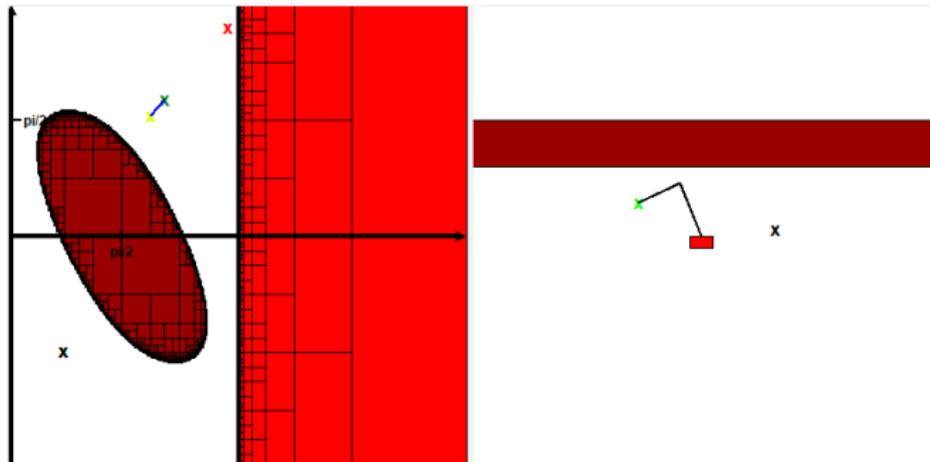


FIGURE – Algorithme du RRT - $n = 2$

Rapidly-exploring random tree (RRT)

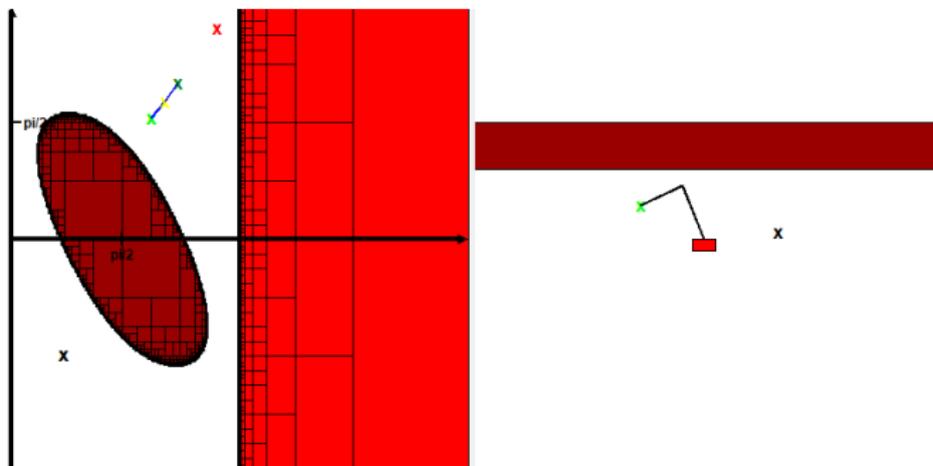


FIGURE – Algorithme du RRT - $n = 3$

Rapidly-exploring random tree (RRT)

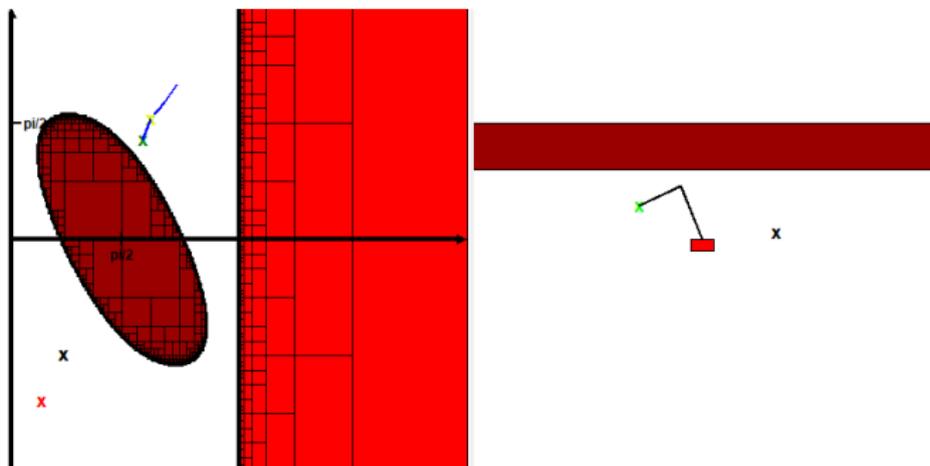


FIGURE – Algorithme du RRT - $n = 4$

Rapidly-exploring random tree (RRT)

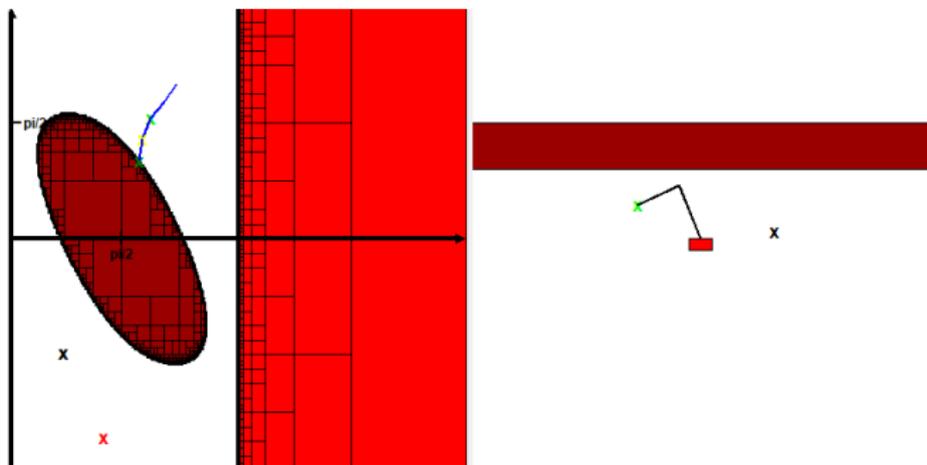


FIGURE – Algorithme du RRT - $n = 5$

Rapidly-exploring random tree (RRT)

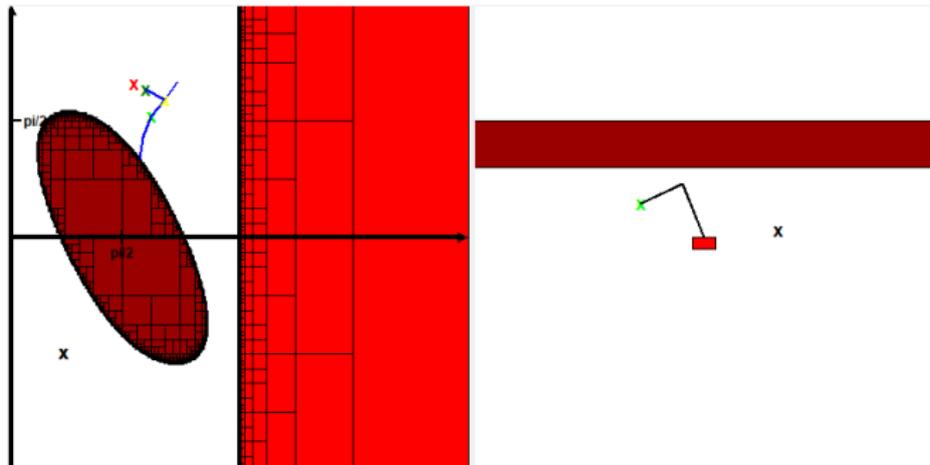


FIGURE – Algorithme du RRT - $n = 6$

Rapidly-exploring random tree (RRT)

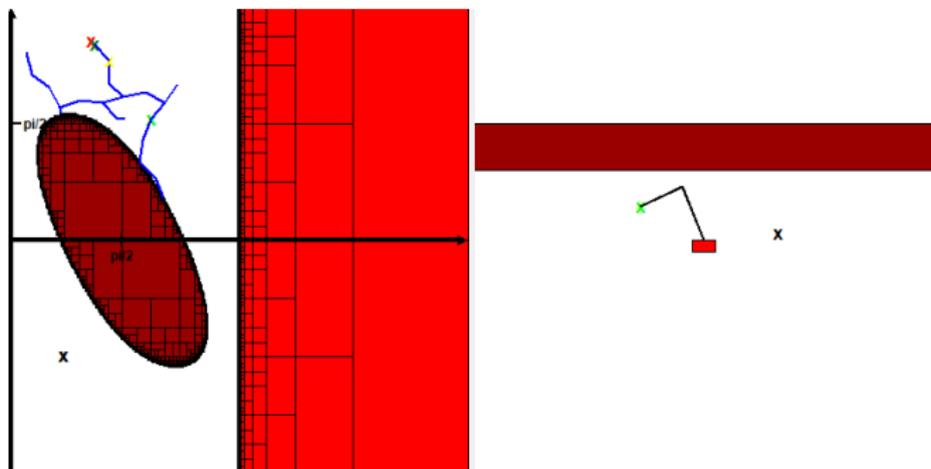


FIGURE – Algorithme du RRT - $n = 20$

Rapidly-exploring random tree (RRT)

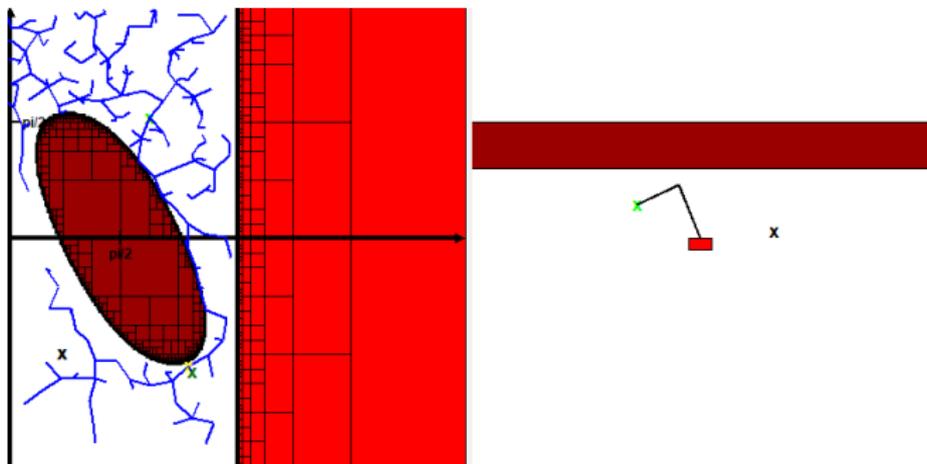


FIGURE – Algorithme du RRT - $n = 200$

Rapidly-exploring random tree (RRT)

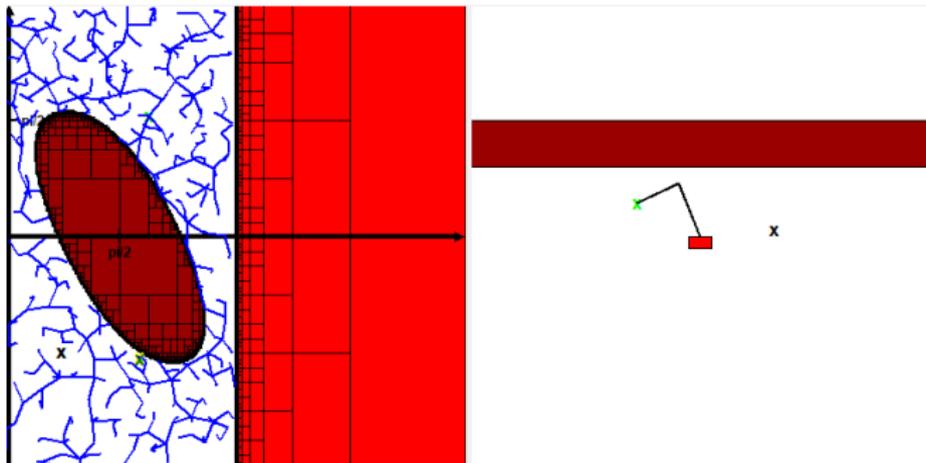


FIGURE – Algorithme du RRT - $n = 500$

Rapidly-exploring random tree (RRT)

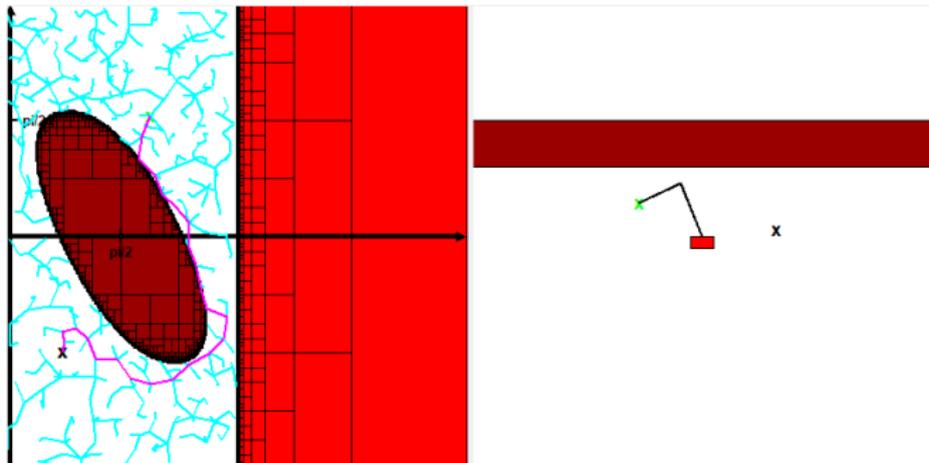


FIGURE – Algorithme du RRT - $n = 500$

Rapidly-exploring random tree (RRT)

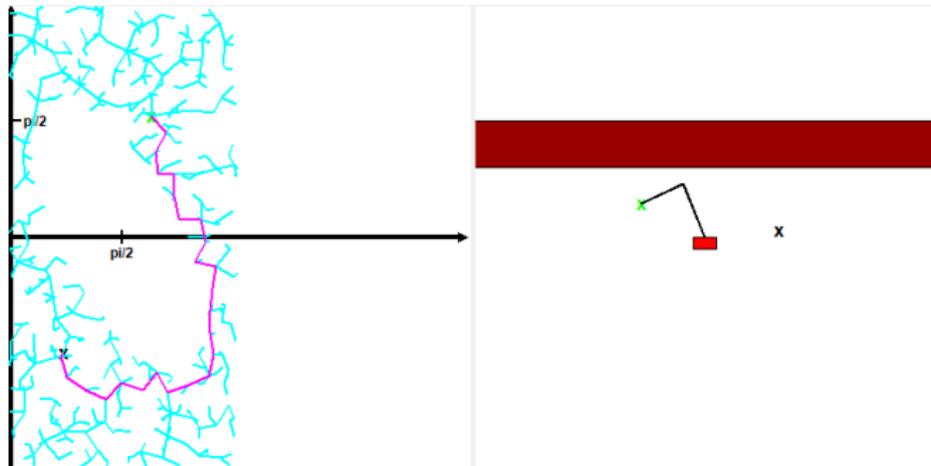


FIGURE – Algorithme du RRT - $n = 500$

Probabilistic RoadMap (PRM)

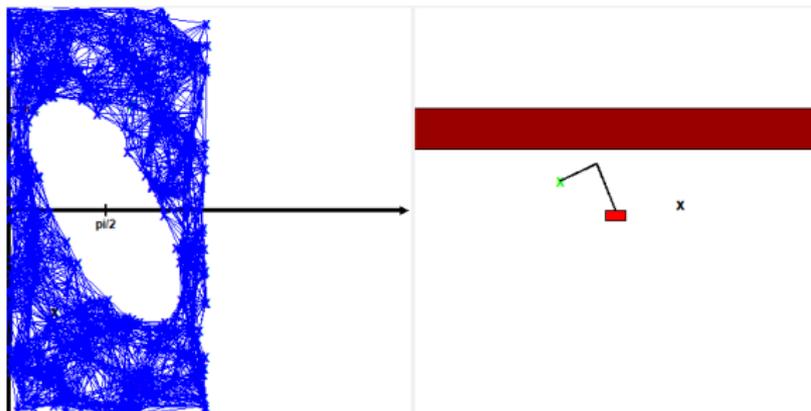
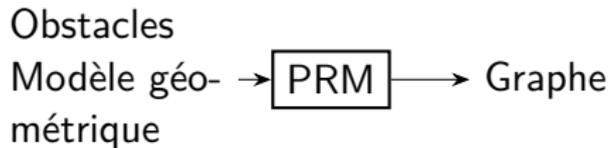


FIGURE – Algorithme du PRM - $n = 500$

Probalistic RoadMap (PRM)

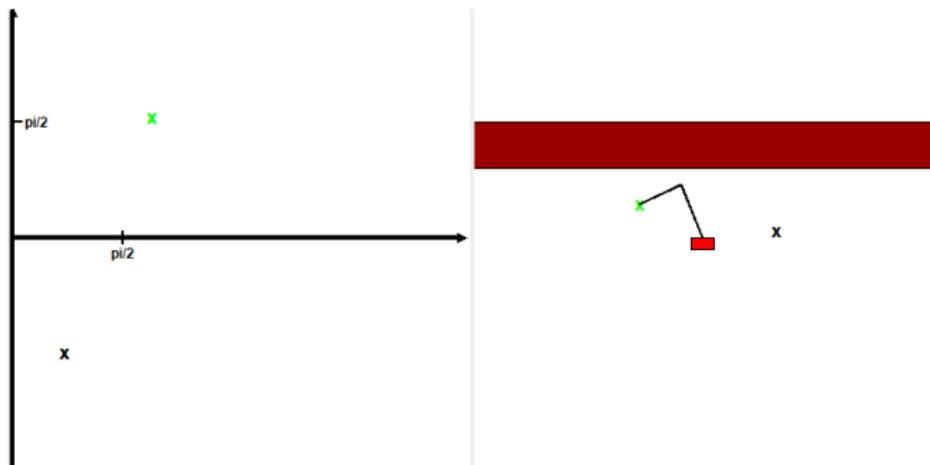


FIGURE – Algorithme du PRM - $n = 0$

Probabilistic RoadMap (PRM)

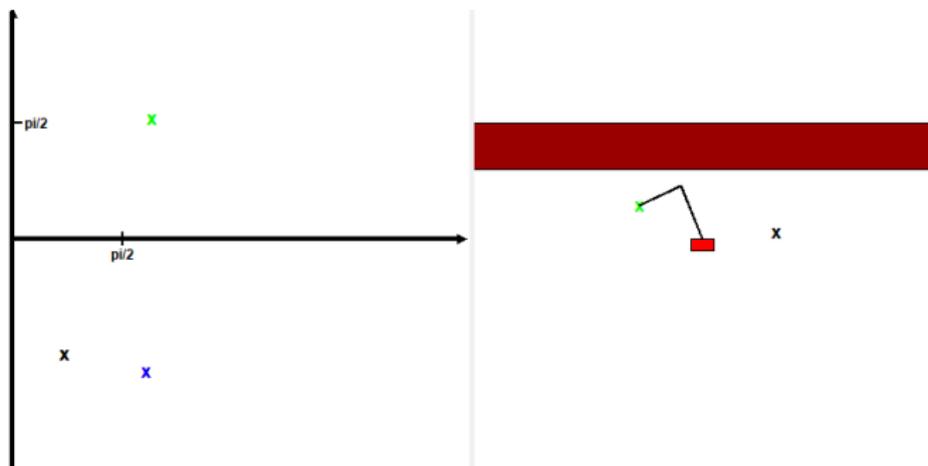


FIGURE – Algorithme du PRM - $n = 1$

Probabilistic RoadMap (PRM)

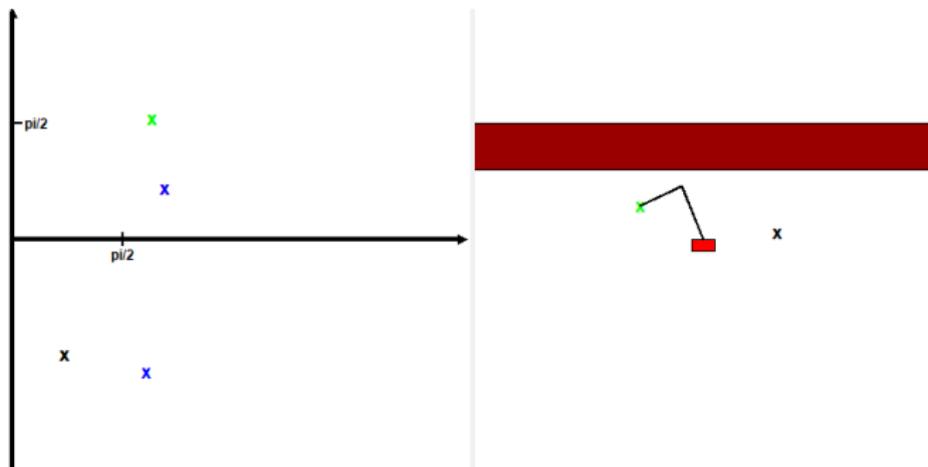


FIGURE – Algorithme du PRM - $n = 2$

Probabilistic RoadMap (PRM)

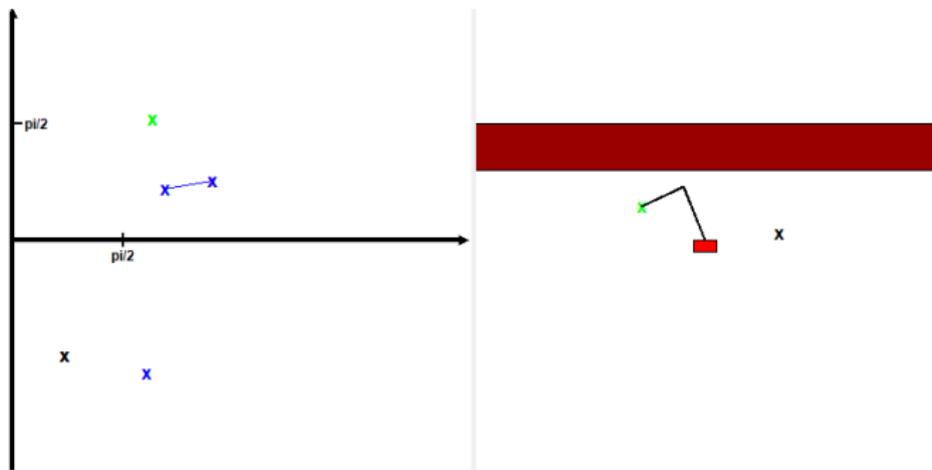


FIGURE – Algorithme du PRM - $n = 3$

Probabilistic RoadMap (PRM)

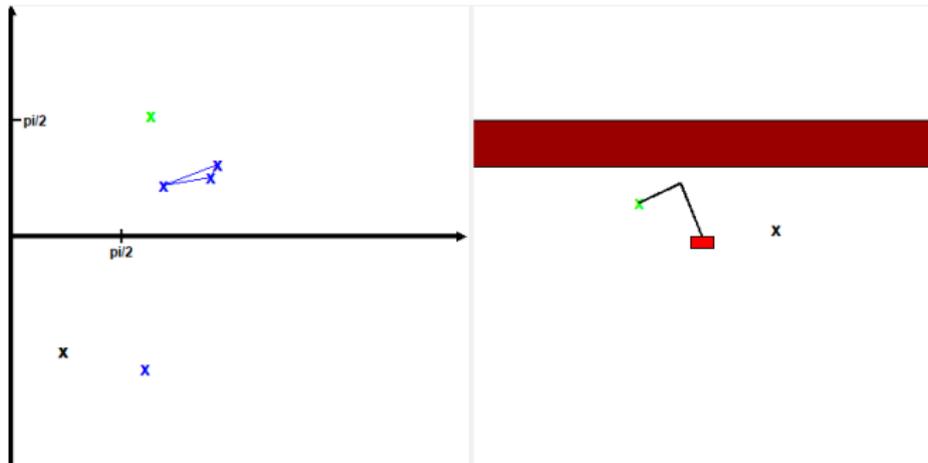


FIGURE – Algorithme du PRM - $n = 4$

Probabilistic RoadMap (PRM)

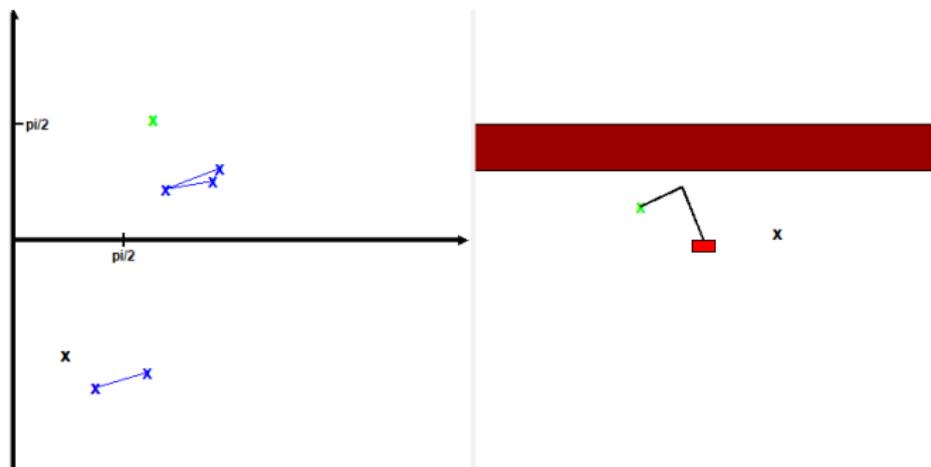


FIGURE – Algorithme du PRM - $n = 5$

Probabilistic RoadMap (PRM)

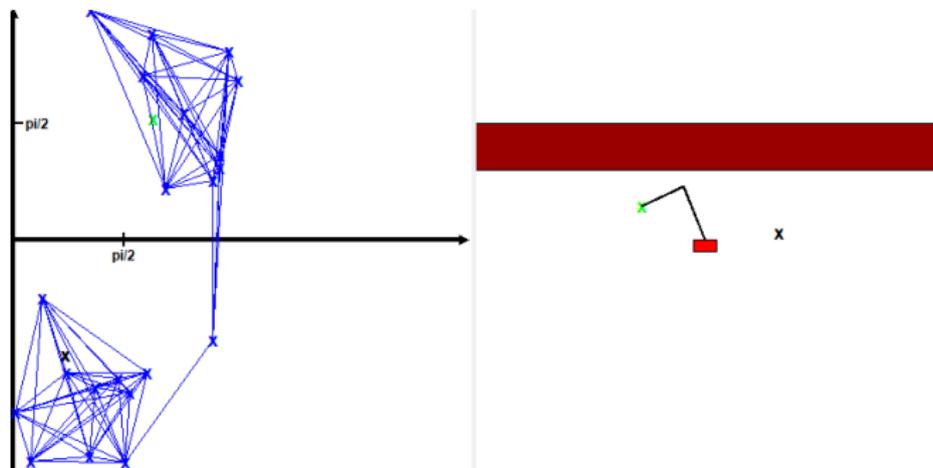


FIGURE – Algorithme du PRM - $n = 20$

Probabilistic RoadMap (PRM)

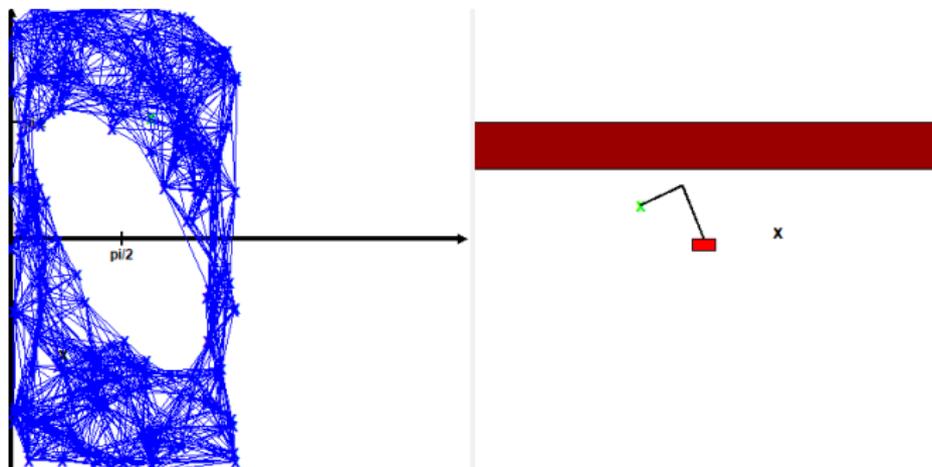


FIGURE – Algorithme du PRM - $n = 200$

Probabilistic RoadMap (PRM)

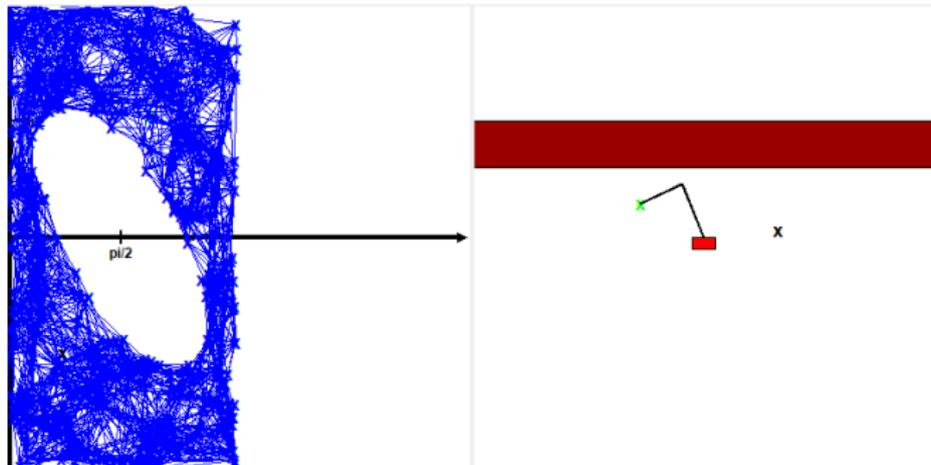


FIGURE – Algorithme du PRM - $n = 500$

Probabilistic RoadMap (PRM)

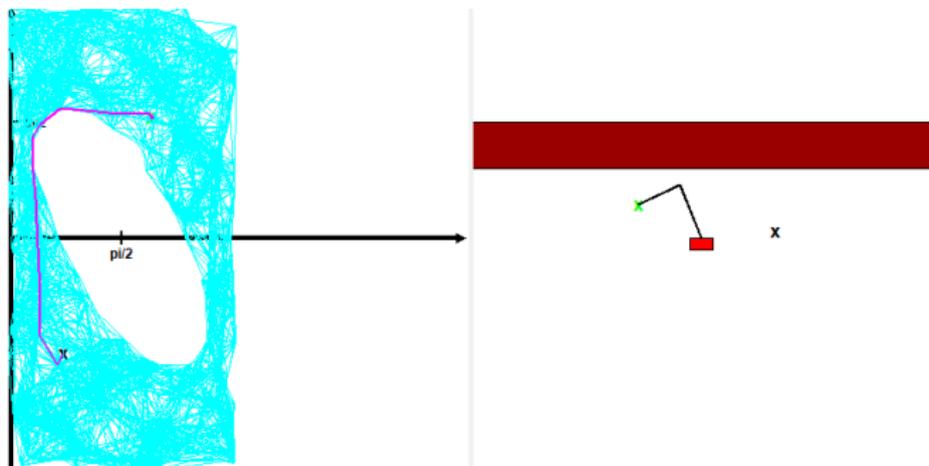


FIGURE – Algorithme du PRM - $n = 500$

Stratégies

Type	Exploration de C_{free}	Décision	Résultat
Topologique	PRM	Connexité	Faisabilité
Géométrique	RRT	TSP	Optimalité

Problème du Voyageur de Commerce

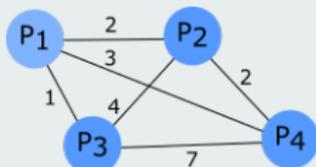
Caractérisation d'un PVC

- n nœuds (points de vue)
- Matrice des coûts $d(i,j)$

Objectif : Trouver la permutation optimale σ^*

$$\sigma^* = \arg \min_{\sigma} \left\{ \sum_{i=1}^{n-1} (d(\sigma_i, \sigma_{i+1})) + d(\sigma_n, \sigma_1) \right\}.$$

Exemple de PVC



$$d_{i,j} = \begin{pmatrix} 0 & 2 & 1 & 3 \\ 2 & 0 & 4 & 2 \\ 1 & 4 & 0 & 7 \\ 3 & 2 & 7 & 0 \end{pmatrix}$$

Problème du Voyageur de Commerce

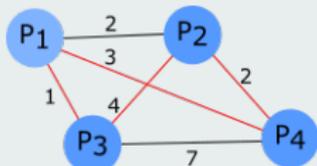
Caractérisation d'un PVC

- n nœuds (points de vue)
- Matrice des coûts $d(i,j)$

Objectif : Trouver la permutation optimale σ^*

$$\sigma^* = \arg \min_{\sigma} \left\{ \sum_{i=1}^{n-1} (d(\sigma_i, \sigma_{i+1})) + d(\sigma_n, \sigma_1) \right\}.$$

Exemple de PVC



$$d_{i,j} = \begin{pmatrix} 0 & 2 & 1 & 3 \\ 2 & 0 & 4 & 2 \\ 1 & 4 & 0 & 7 \\ 3 & 2 & 7 & 0 \end{pmatrix}$$

Algorithmes de résolution

Type	Exact		Heuristique			
Nom	Bruteforce	Held-Karp	Plus proche voisin	Christofides	2-opt	3-opt
Complexité	$\mathcal{O}(n!)$	$\mathcal{O}(n^2 2^n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$		
Facteur d'approximation	1	1	$\lceil \frac{1}{2} \ln(n) \rceil + \frac{1}{2}$	1.5		

TABLE – Plusieurs algorithmes de résolution

Connexité

Définition

Pour un graphe $G = (V, E)$, (v_1, v_2) connexe
 $\iff \exists \gamma$ tel que $\gamma(0) = v_1$ et $\gamma(1) = v_2$.

Définition

Un graphe G est dit connexe $\iff \forall (v_i, v_j) \in E, (v_i, v_j)$ est connexe

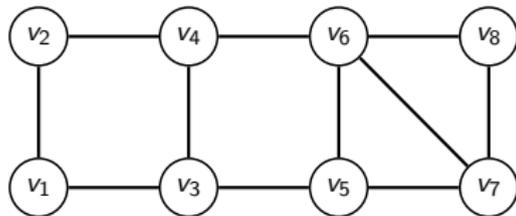


FIGURE – Exemple de graphe connexe

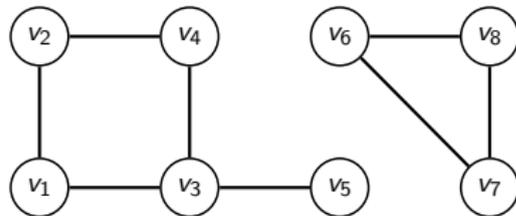


FIGURE – Exemple de graphe non connexe

Plusieurs solutions au modèle géométrique inverse

Modèle Géométrique

Direct :

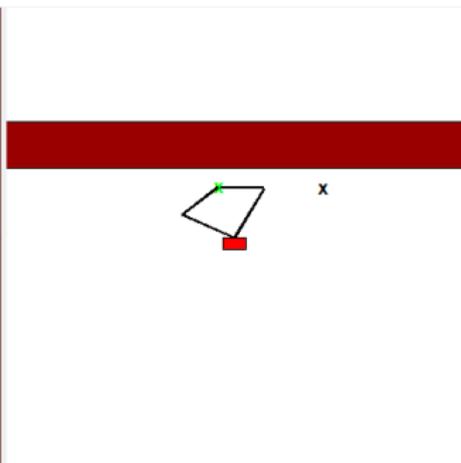
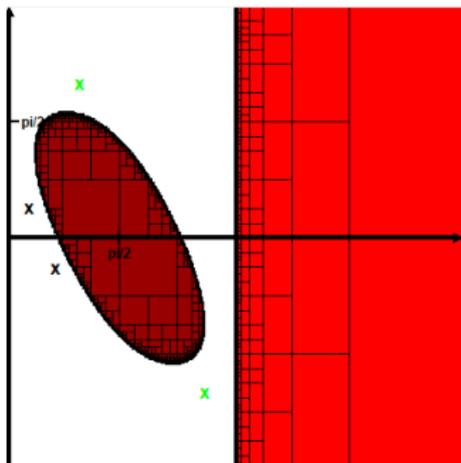
$$f: \mathbf{Q} \rightarrow \mathbf{W}$$

$$f(q) \mapsto x$$

Inverse :

$$f^{-1}: \mathbf{W} \rightarrow 2^{\mathbf{Q}}$$

$$f^{-1}(x) \mapsto \{q\}$$



Extension de la connexité

Définition

Étant donné un graphe $G = (V, E)$ et $\mathcal{V} = \{V_1, V_2, \dots, V_k\}$ une famille de sous-ensembles de V on dit que G est \mathcal{V} -connexe s'il existe un chemin α visitant tout les sous-ensembles V_i de \mathcal{V} .

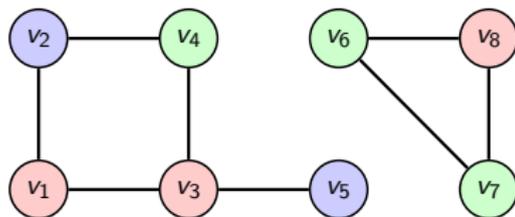


FIGURE – Exemple de graphe \mathcal{V} -connexe

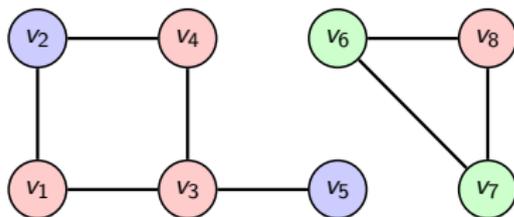


FIGURE – Exemple de graphe non \mathcal{V} -connexe

Merci pour votre attention.