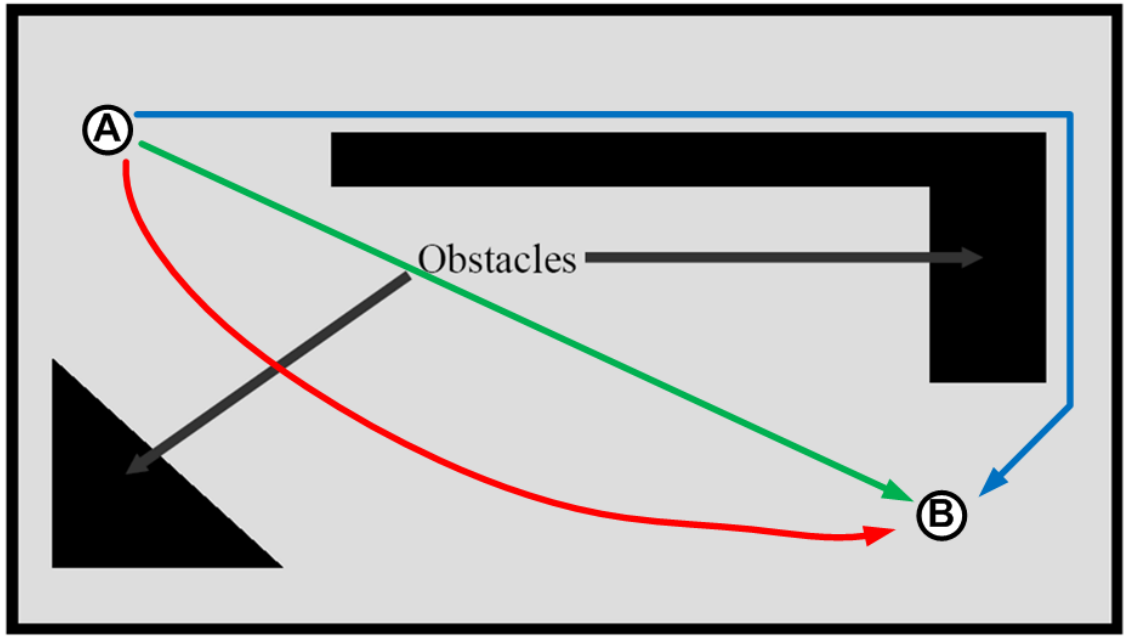
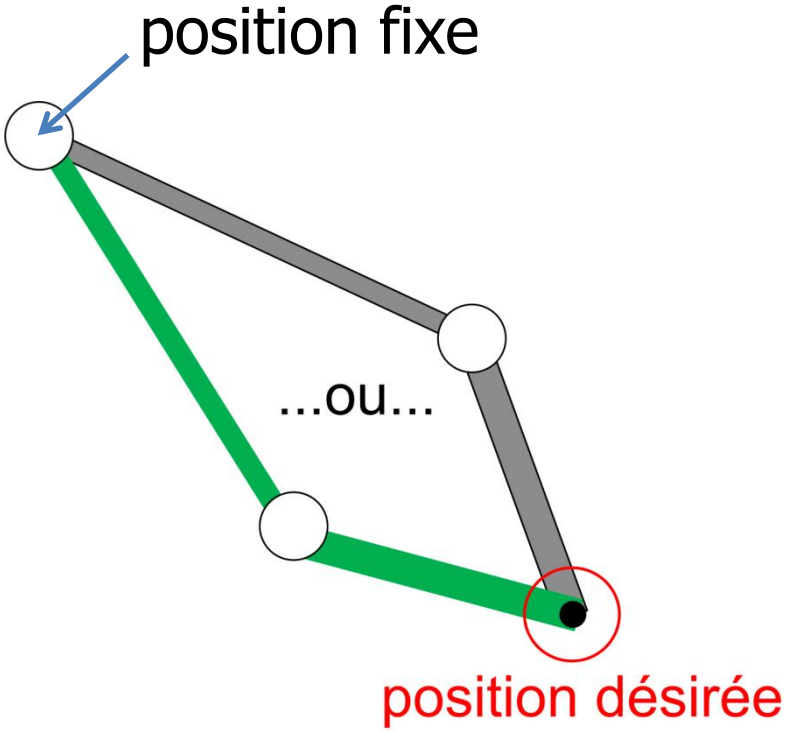
Planification

manuel section 6.2 et 6.3, p.176-208

Planification : mal posé

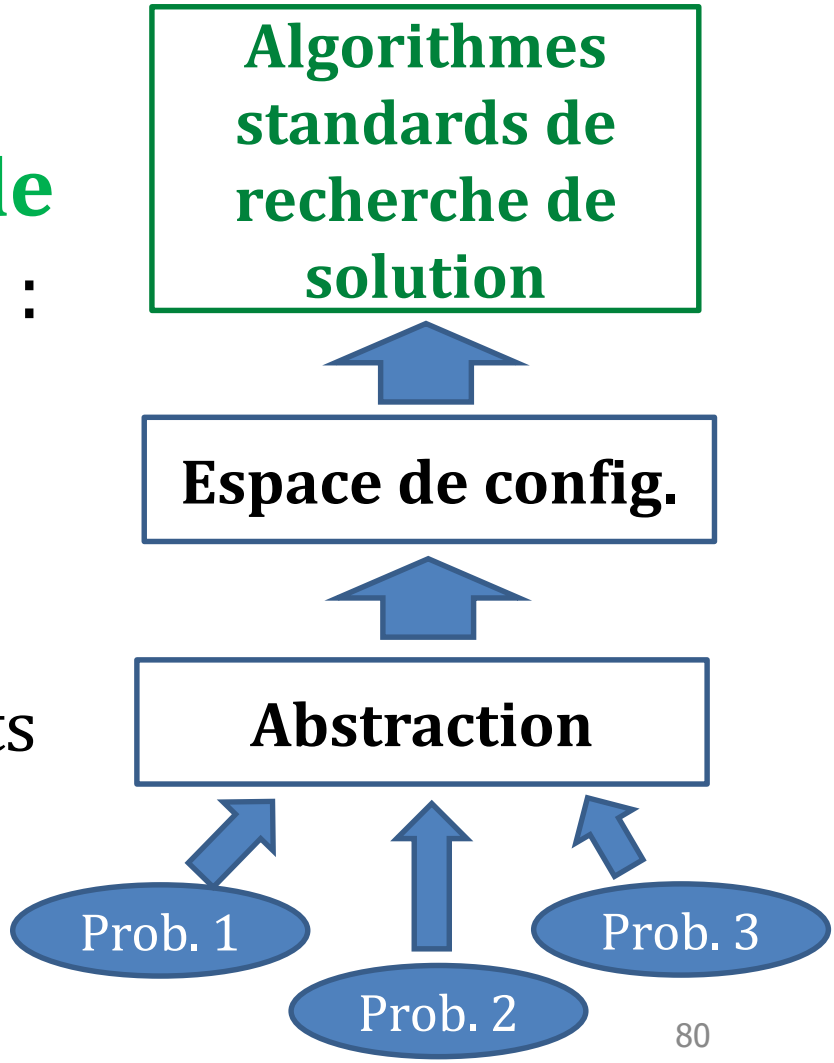
- Déplacer un robot de A vers B
- Positionner l'extrémité d'un bras



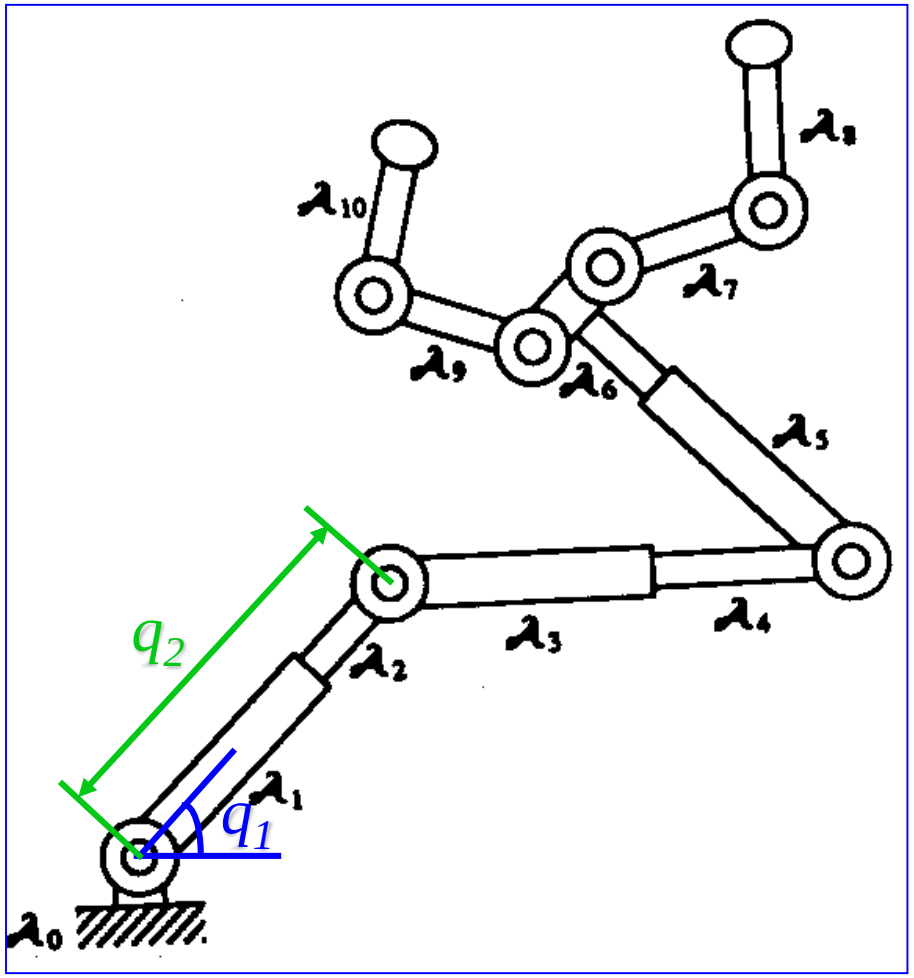


Espace de configuration

- Représentation abstraite d'un problème de planification
- Employer des **algorithmes de recherches** génériques pour :
  - robot 2D glisse en x, y
  - robot conduite différentielle
  - bras articulé
  - robot humanoïde avec 30 joints
  - etc.



Exemple configuration (bras articulé)

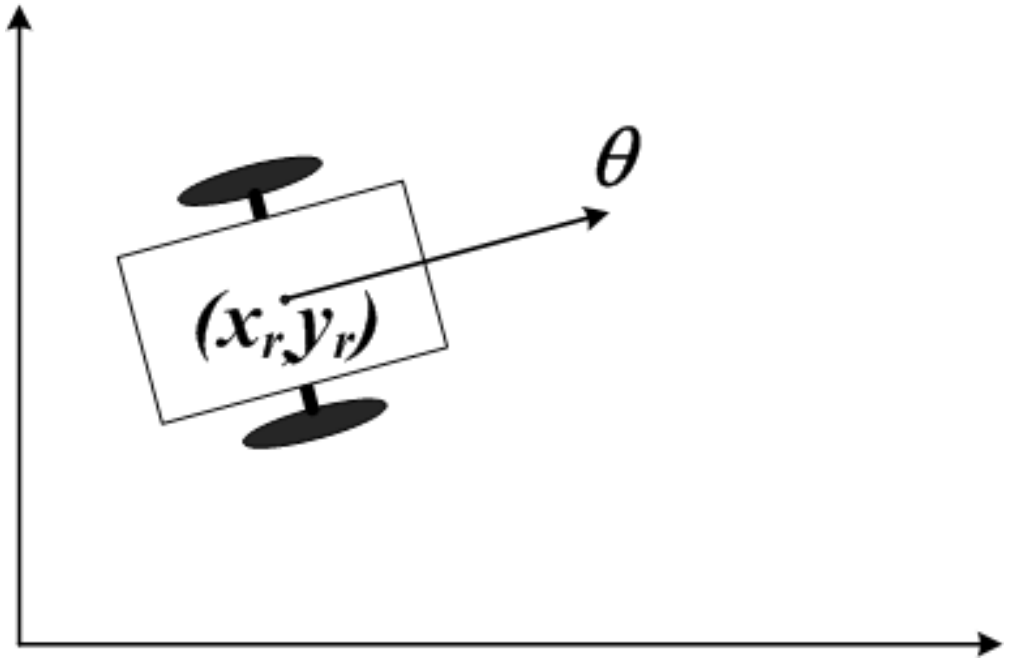


- La configuration d'un bras est l'ensemble des angles θ_i et étirements d_j.
- Pour un robot articulé complexe de 10 actionneurs :

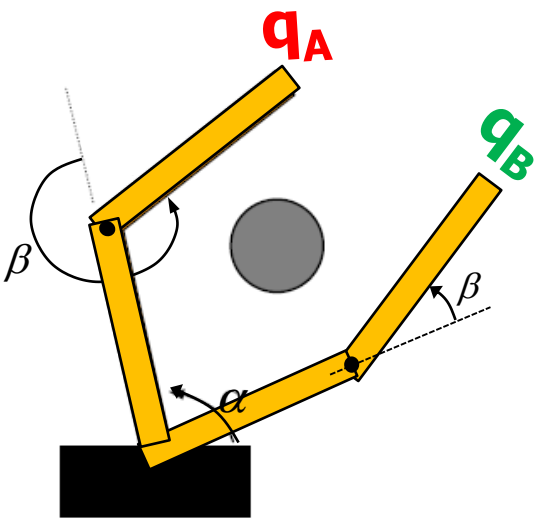
$$\vec{q} = (q_1, q_2, \cdots, q_{10})$$

Exemple configuration (robot mobile)

- Espace de travail en 2D
- Configuration : $\vec{q} = (x_r, y_r, \theta)$

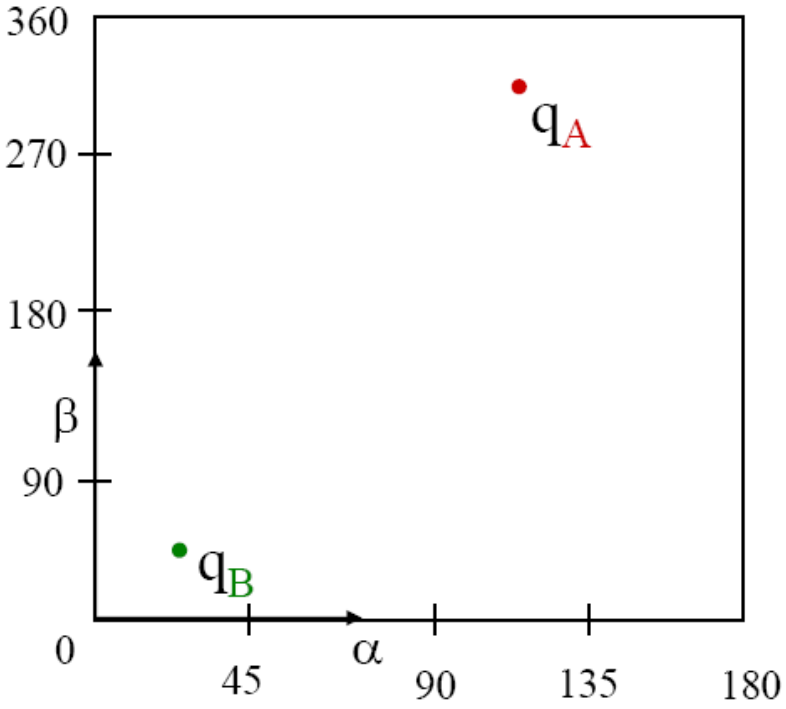


Obstacle dans l'espace de travail



Un obstacle dans l'espace de travail

Où placer  ?

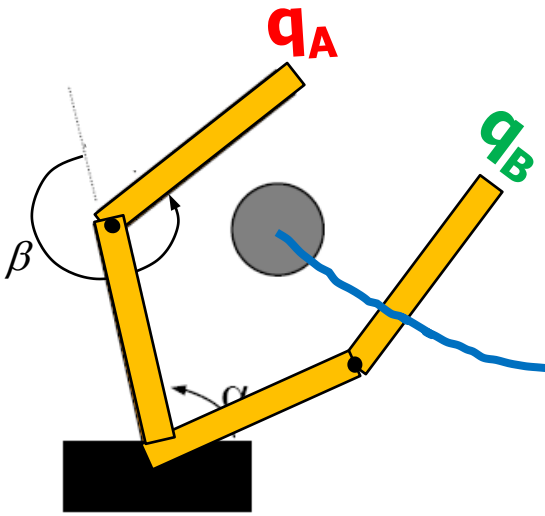


Espace de configuration

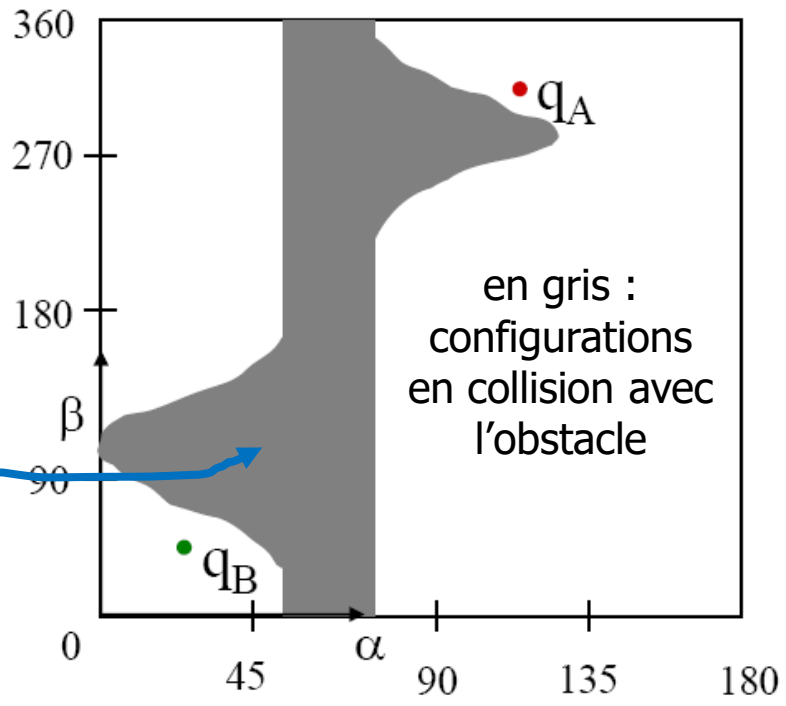
(en 2D, car 2 joints)

Obstacle dans l'espace des configurations

Comment aller de q_A vers q_B?



Un obstacle dans l'espace de travail

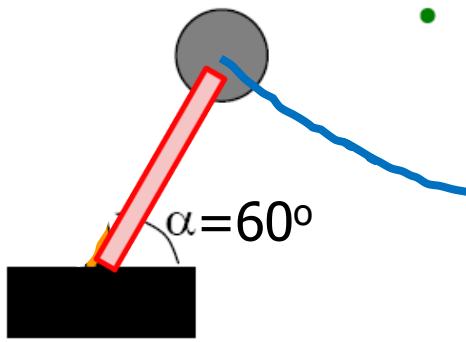


Espace de configuration

(en 2D, car 2 joints)

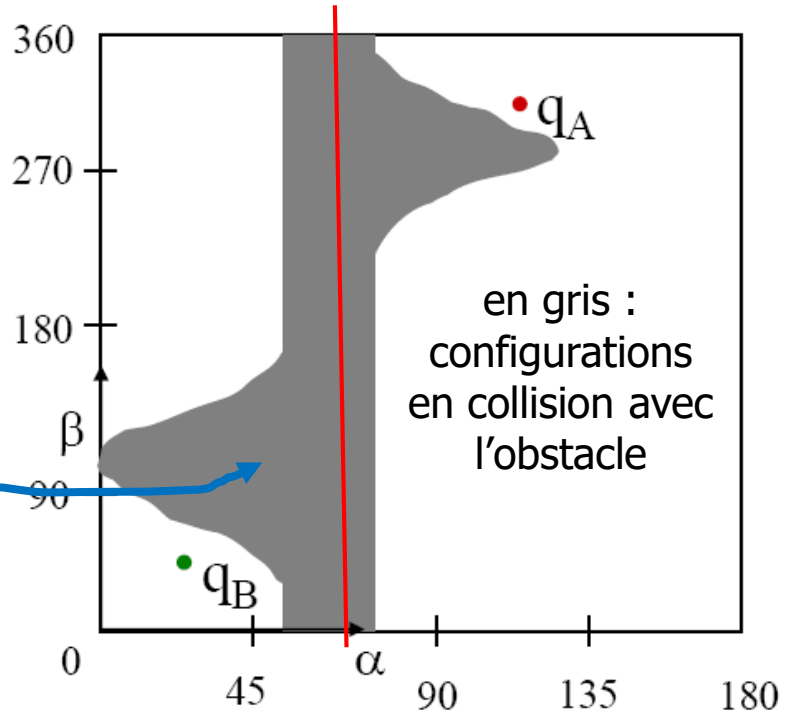
Obstacle dans l'espace des configurations

dès que $\alpha = 60^o$, on a collision, peu importe β



Un obstacle dans l'espace de travail

Comment aller de q_A vers q_B?



Espace de configuration

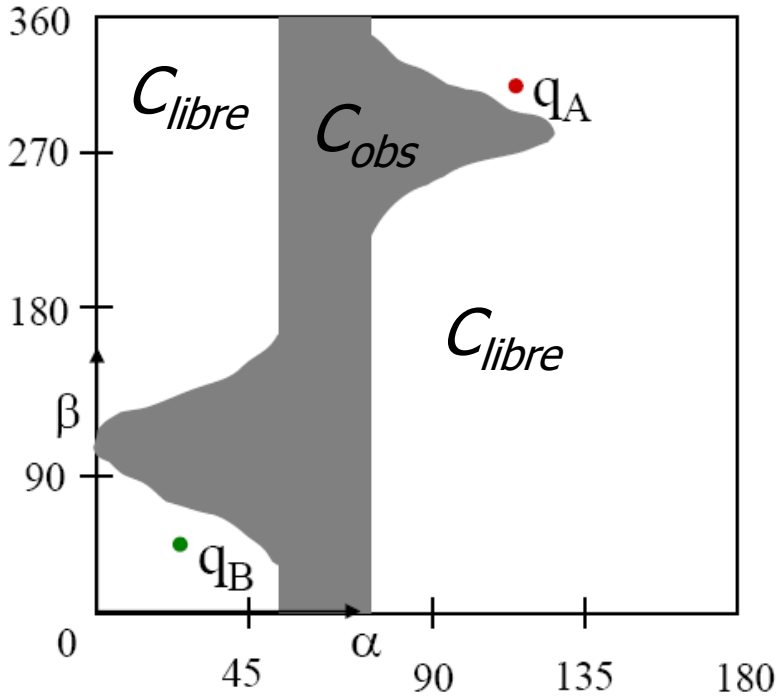
(en 2D, car 2 joints)

Espace des configurations : terminologie

L'espace des configurations C est composé :

- espace libre C_{libre}
- espace des obstacles C_{obs}

$$C = C_{libre} + C_{obs}$$

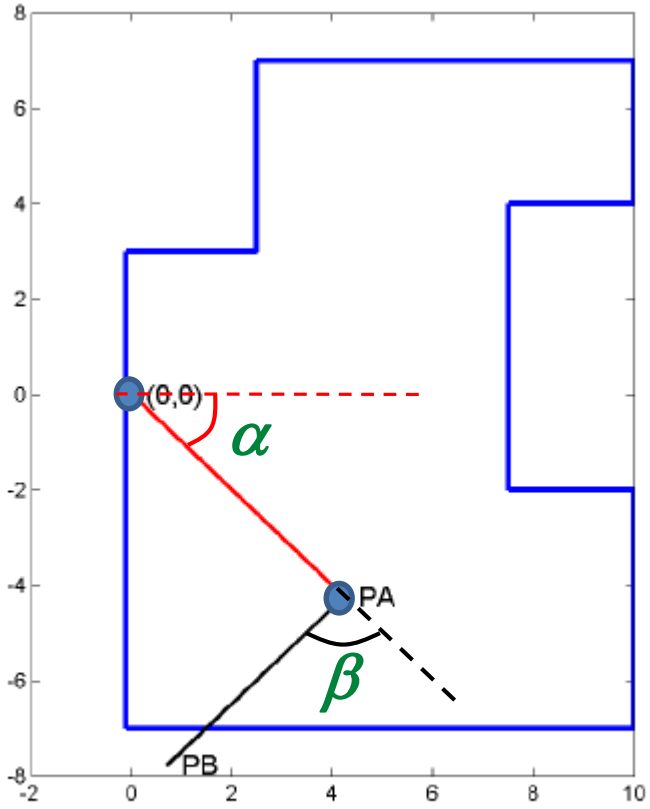


Espace de configuration

(en 2D, car 2 joints)

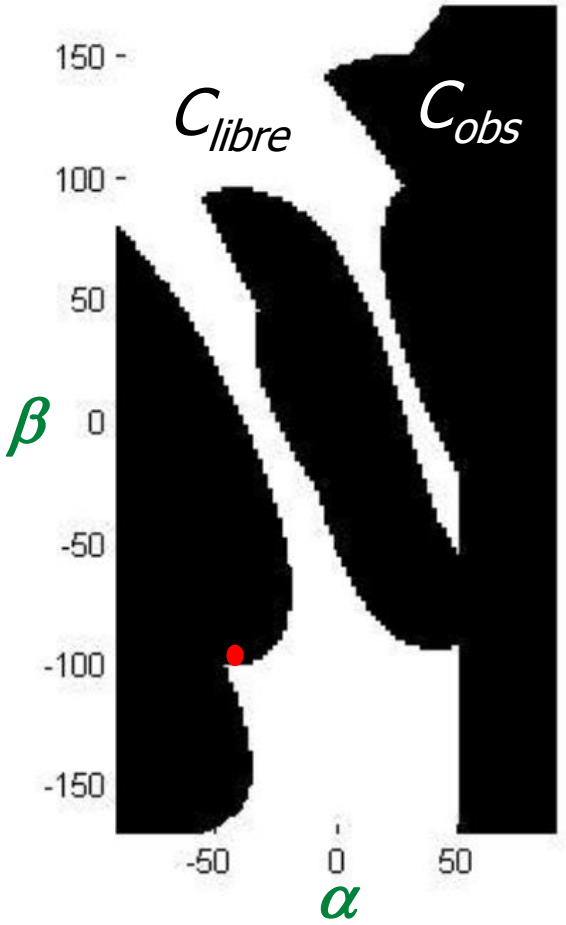
Obstacle dans l'espace des configurations

- Tiré du TP de 2011



$\alpha = -45^o, \beta = -90^o$

espace des configurations



Obstacle dans l'espace des configurations

construction de l'espace des configurations avec obstacles

C_{libre}

C_{obs}

Obstacle dans l'espace des configurations

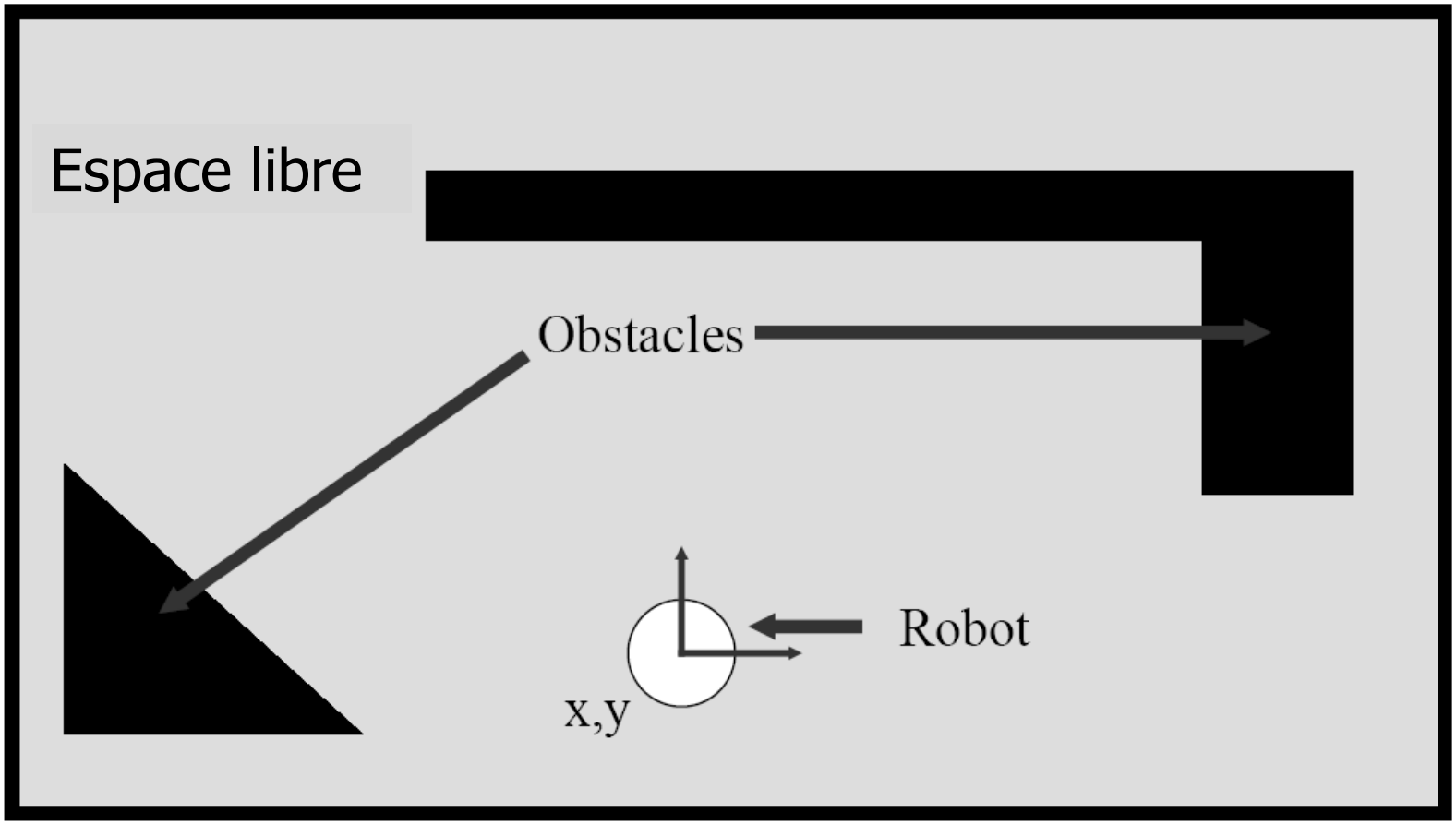
construction de l'espace des configurations avec obstacles

C_{libre}

C_{obs}

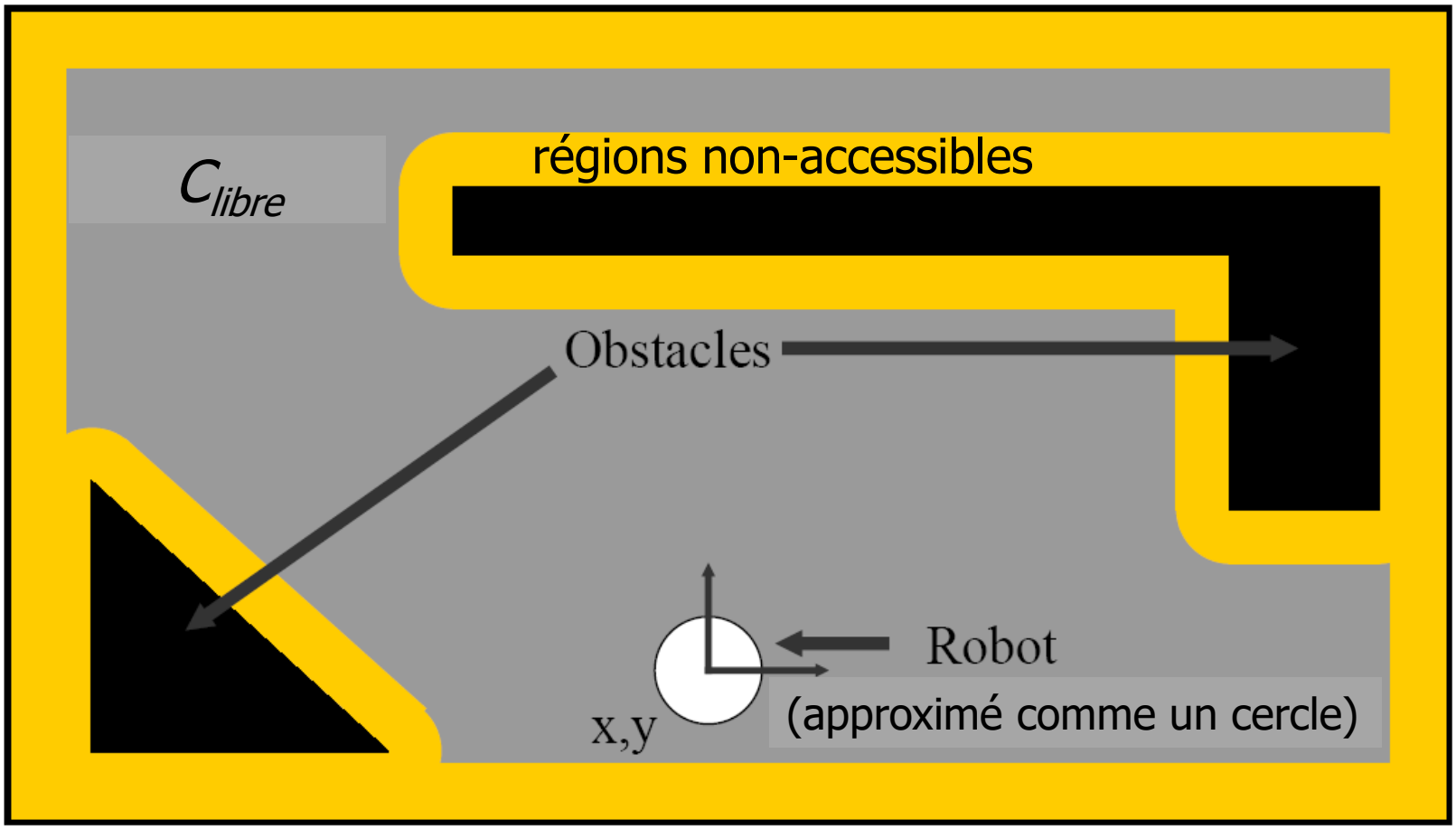
Espace de configuration C

- Pour un robot circulaire se déplaçant en x-y



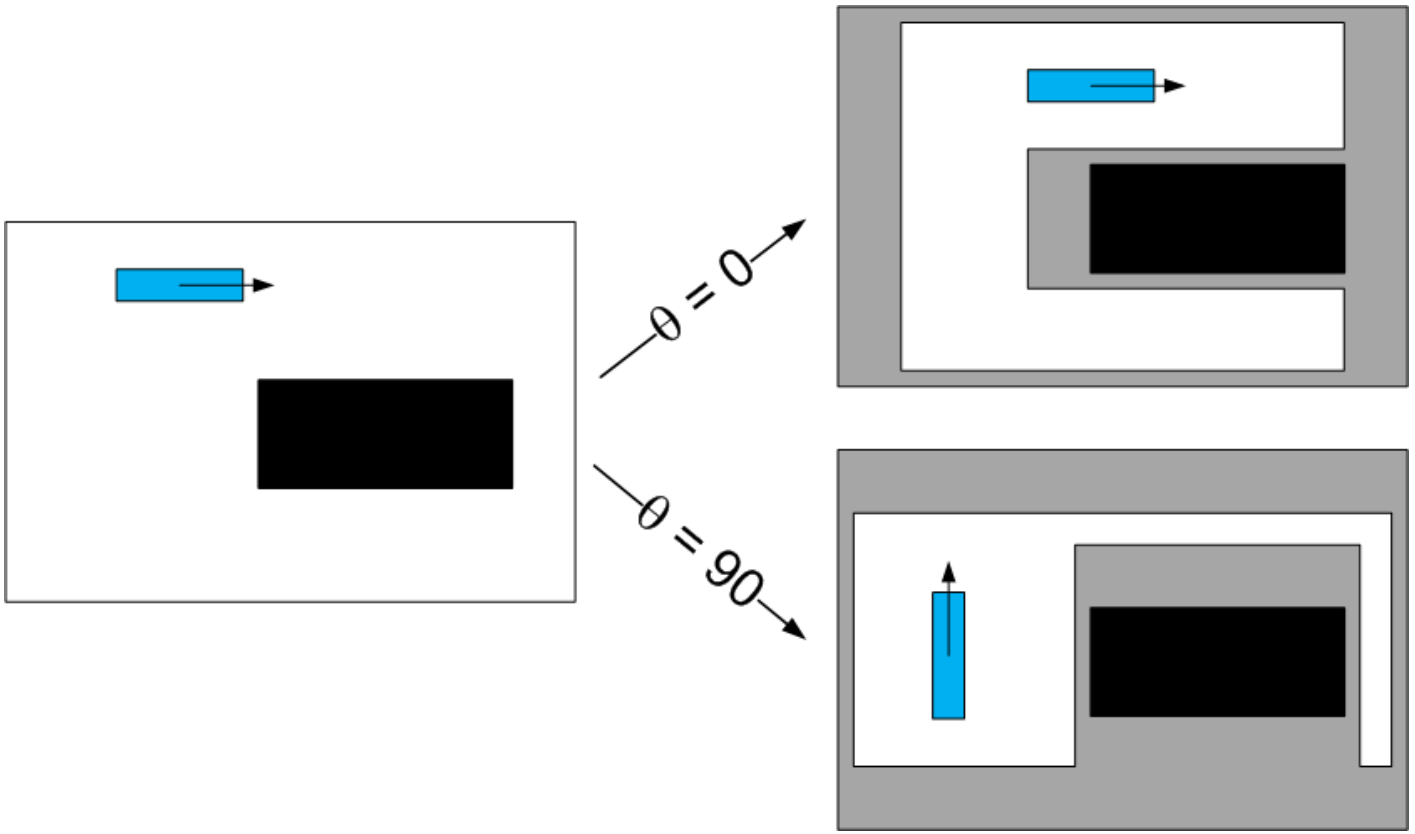
Espace de configuration C

- Le centre du robot ne peut être que dans le gris



Espace de configuration C

- Pour un robot se déplaçant en conduite différentielle : (x, y, θ) (3 dimensions)



Planification de trajectoire : abstraction

- …aussi appelé « ***piano mover's problem*** »
- On a :
  - espace de travail $\boldsymbol{W}$ en $\mathbb{R}^2$ ou $\mathbb{R}^3$
  - régions d'obstacles $\boldsymbol{O}$
  - un robot $\boldsymbol{R}$ avec géométrie dans $\boldsymbol{W}$

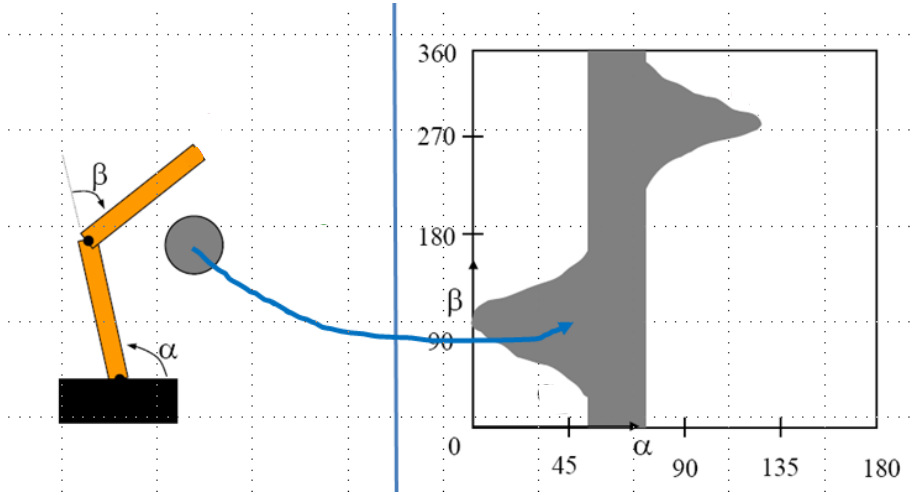
  } **« monde physique »**

  - espace configuration $\boldsymbol{C}$ (avec $\boldsymbol{C} = \boldsymbol{C_{libre}} + \boldsymbol{C_{obs}}$)
  - configuration initiale $q_1 \in \boldsymbol{C_{libre}}$
  - configuration finale $q_2 \in \boldsymbol{C_{libre}}$
- Trouver une trajectoire $\boldsymbol{\tau}$ <u>continue</u> dans $\boldsymbol{C_{libre}}$ entre q_1 et q_2.

Planification de trajectoire : abstraction

- …aussi appelé « ***piano mover's problem*** »
- On a :
  - espace de travail $\boldsymbol{W}$ en $\mathbb{R}^2$ ou $\mathbb{R}^3$
  - régions d'obstacles $\boldsymbol{O}$
  - un robot $\boldsymbol{R}$ avec géométrie dans $\boldsymbol{W}$
  - espace configuration $\boldsymbol{C}$ (avec $\boldsymbol{C} = \boldsymbol{C_{libre}} + \boldsymbol{C_{obs}}$)

calcul de $\boldsymbol{C_{obs}}$ n'est pas facile



Planification de trajectoire : abstraction

- …aussi appelé « ***piano mover's problem*** »
- On a :
  - espace de travail $\boldsymbol{W}$ en $\mathbb{R}^2$ ou $\mathbb{R}^3$
  - régions d'obstacles $\boldsymbol{O}$
  - un robot $\boldsymbol{R}$ avec géométrie dans $\boldsymbol{W}$
  - espace configuration $\boldsymbol{C}$ (avec $\boldsymbol{C} = \boldsymbol{C_{libre}} + \boldsymbol{C_{obs}}$)
  - configuration initiale $q_1 \in \boldsymbol{C_{libre}}$
  - configuration finale $q_2 \in \boldsymbol{C_{libre}}$

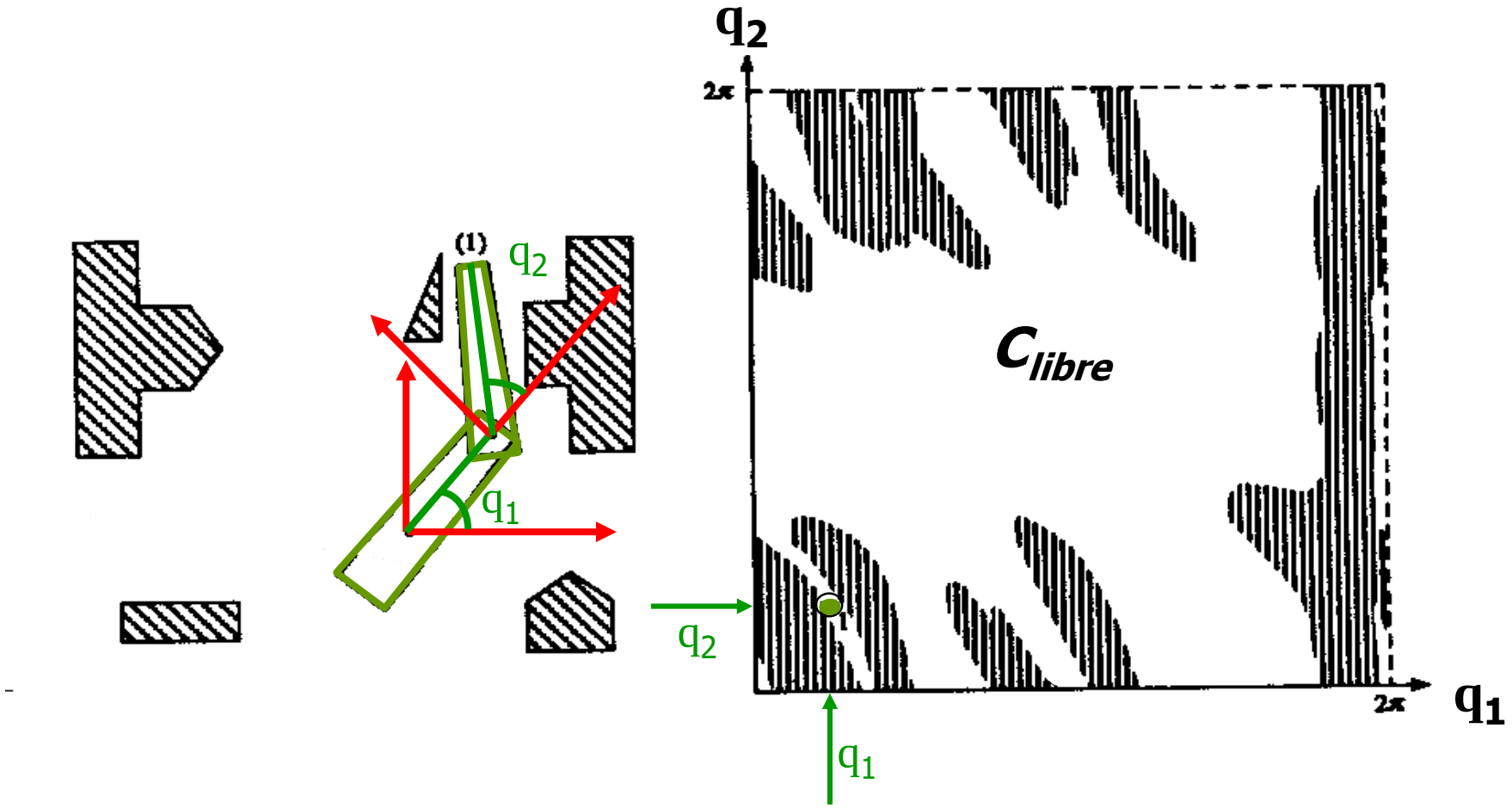
calcul de $\boldsymbol{C_{obs}}$ n'est pas facile

- Trouver une trajectoire τ <u>continue</u> dans $\boldsymbol{C_{libre}}$ entre q_1 et q_2.

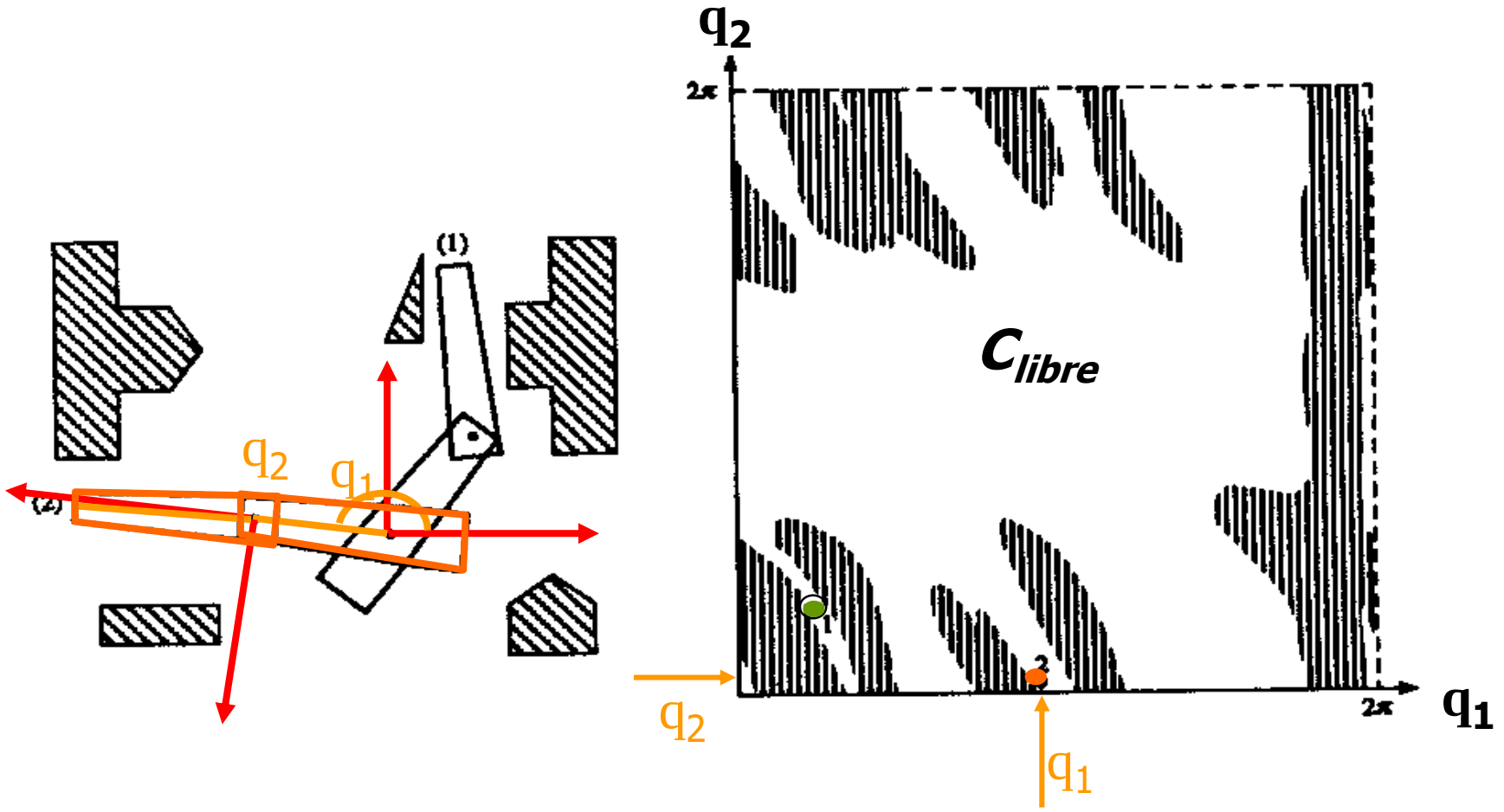
calcul exact approximativement exponentiel en dimension de q

Espace de configuration : position 1

- Pour un bras avec 2 joints rotatifs q_1 et q_2

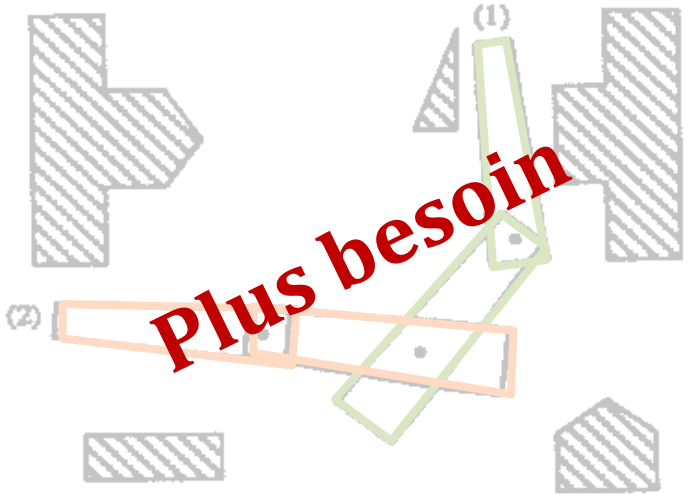


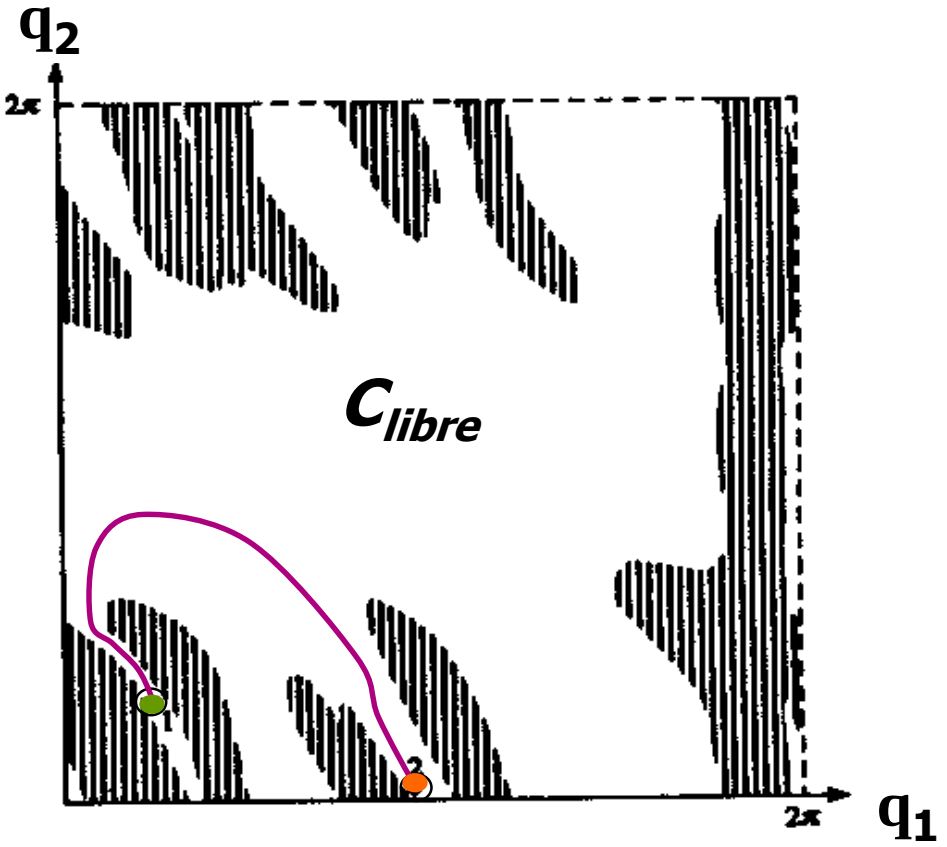
Espace de configuration : position 2



Espace de configuration

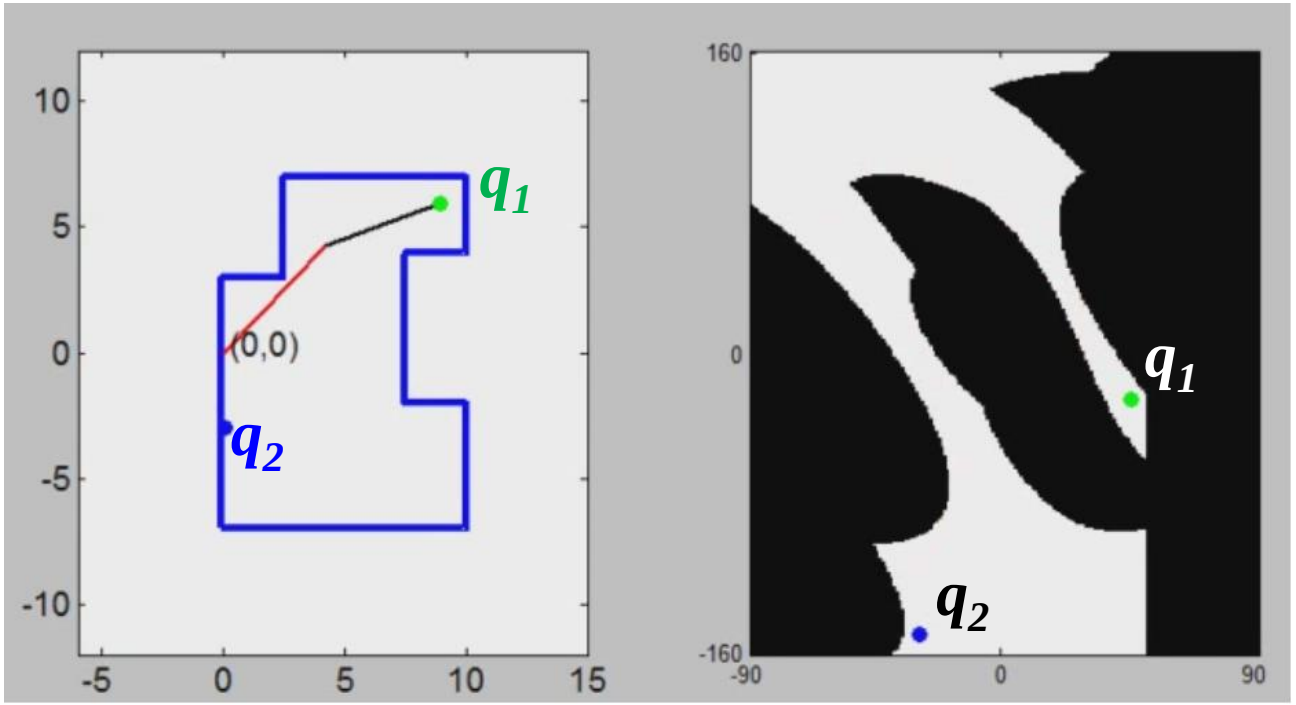
- Trajectoire possible entre q_1 et q_2 ? **oui!**



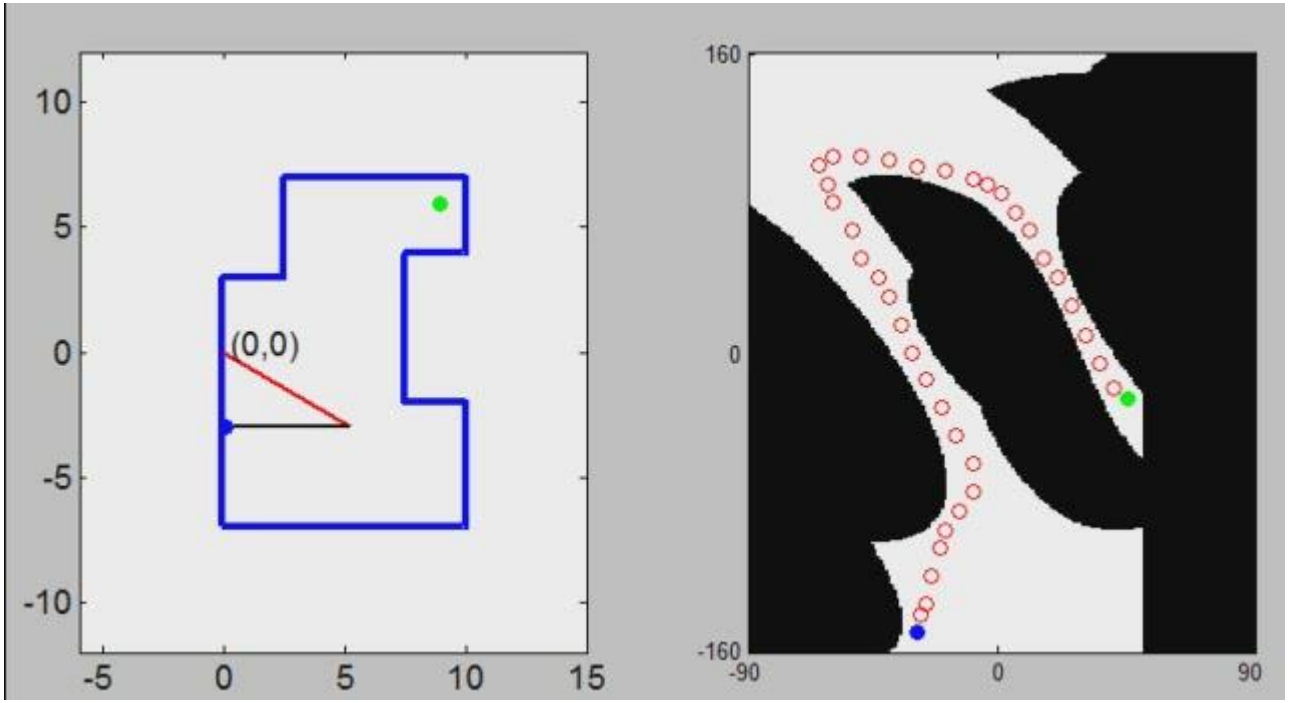


Exemple : bras robotisé

On veut déplacer le bout du bras *(end effector)* de q_1 à q_2.

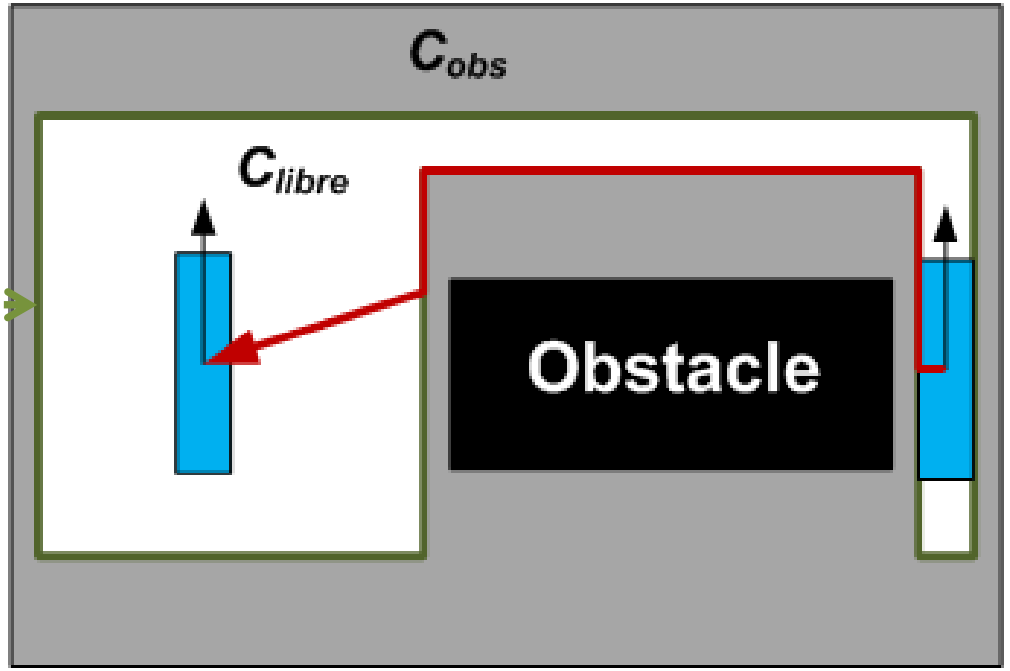


Exemple : bras robotisé



Configuration semi-libre

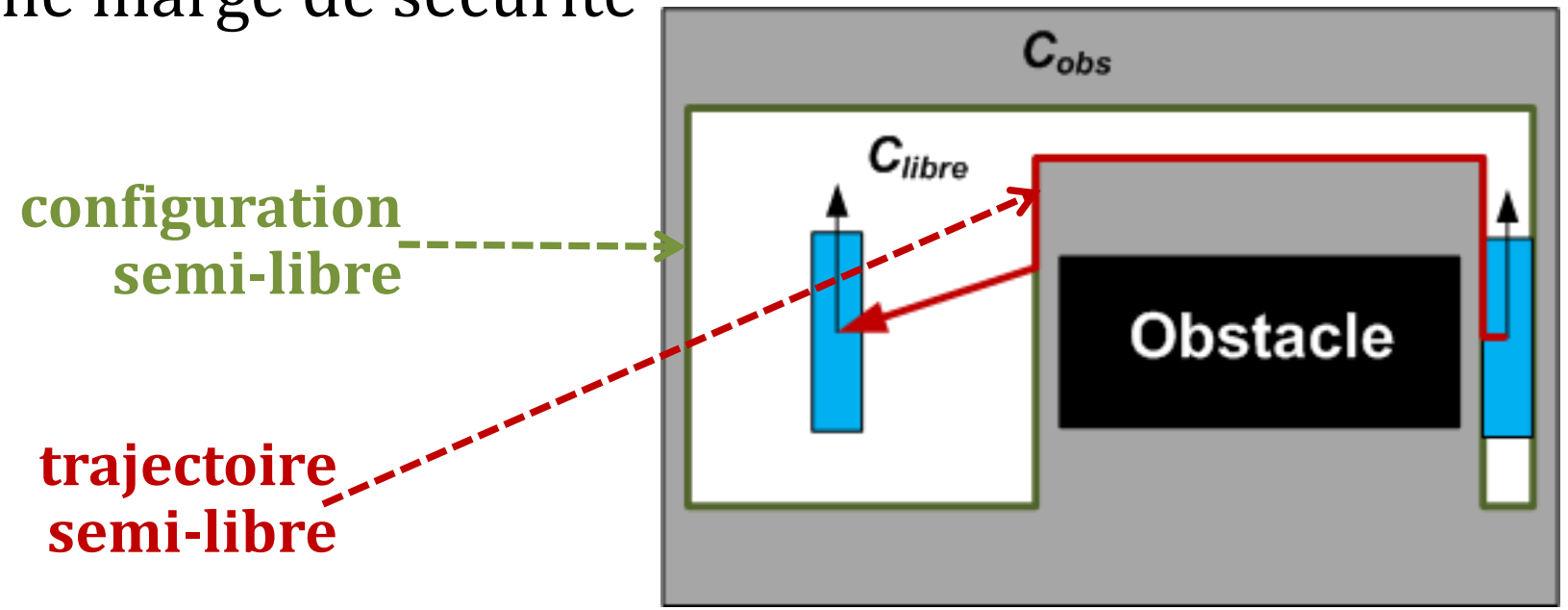
- Une configuration est <u>semi-libre</u> si le robot touche un obstacle, sans le pénétrer.
- Les configurations semi-libres sont à la bordure entre C_{obs} et C_{libre}.



configuration semi-libre

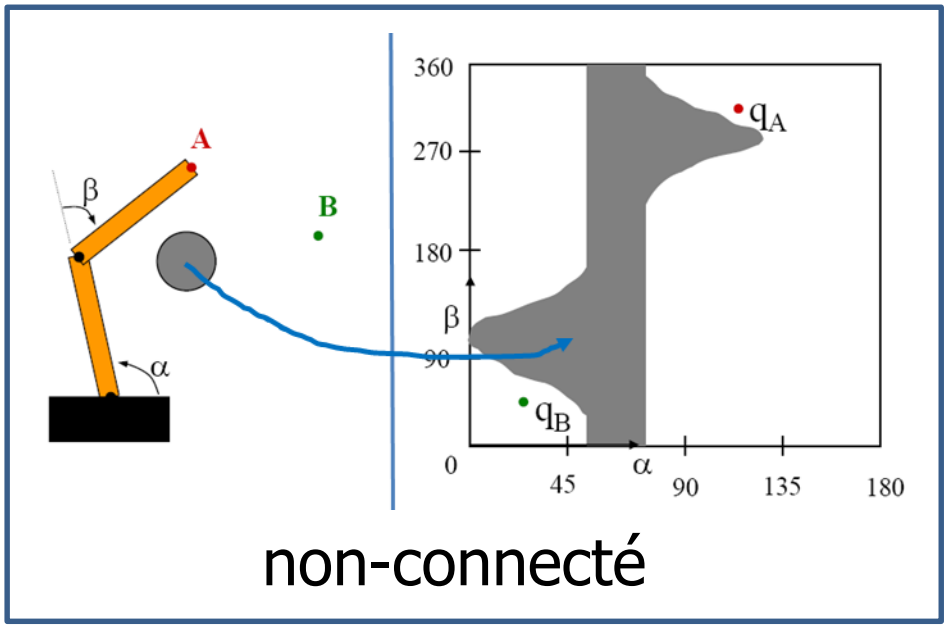
Trajectoire semi-libre

- Trajectoire est libre si entièrement dans C_{libre}.
- Trajectoire est semi-libre si elle contient au moins une configuration semi-libre.
- On cherche, en général, à éviter ces configurations : aucune marge de sécurité

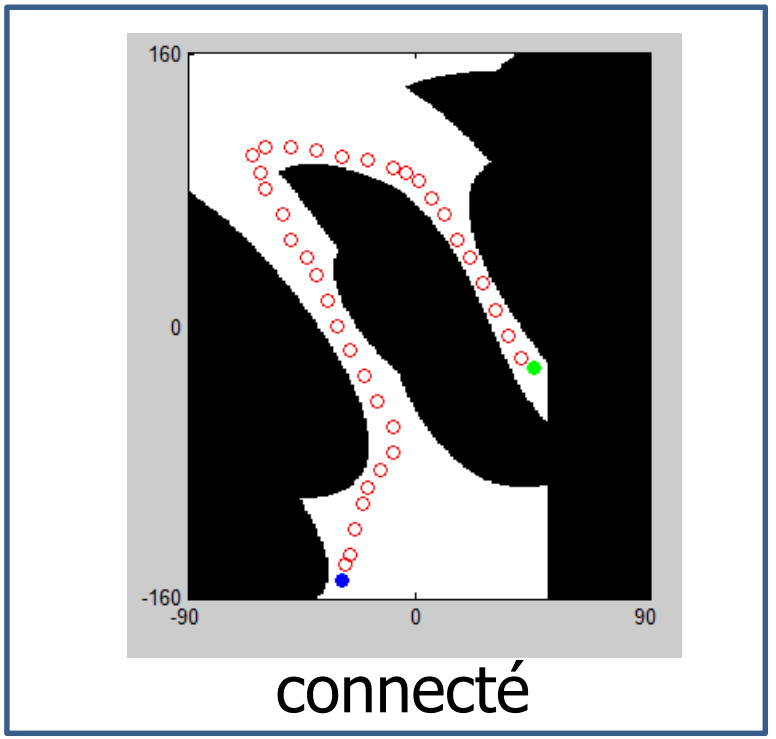


Connectivité de l'espace-C

- L'espace de configuration $\boldsymbol{C}$ est <u>connecté</u> s'il existe une trajectoire entre n'importe quelle paire de configuration q_1 et q_2 dans $\boldsymbol{C_{libre}}$.



non-connecté



connecté

Planification de mouvements

- Cherche à trouver une trajectoire dans l'espace des configurations, sans collision → C_{libre}
- **Méthodes déterministes**
  - à chaque fois, l'algorithme retourne la même réponse
- **Probabilistes**
  - fonctionne par échantillonnage au hasard
  - ne trouve pas toujours la même trajectoire

Méthodes Déterministes

- Par discrétisation :
  - Graphe des visibilités
  - Décomposition cellulaire verticale
  - ~~Diagrammes de Voronoï~~
- ~~Algorithmes *bug0*, *bug1* et *bug2*~~
- Champs de potentiels
- Propagation de front d'ondes

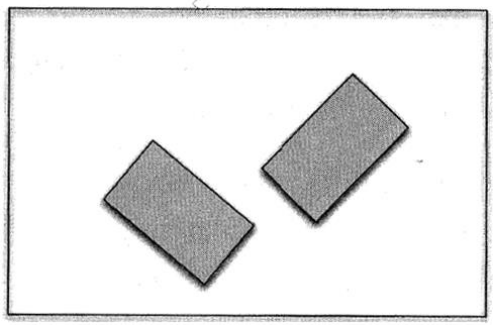
Solutions exactes vs approximatives

- Un algorithme **exact** :
  - trouvera une solution si elle existe

    → **algorithme complet**
  - temps de calcul trop important pour des problèmes intéressants (e.g. humanoïdes à 30 joints).
- Un algorithme **approximatif** :
  - n'assure pas de trouver la solution…
  - …mais est plus rapide.
  - *e.g.* grille discrète des configurations
    - (angle multiples de 5º, par exemple)

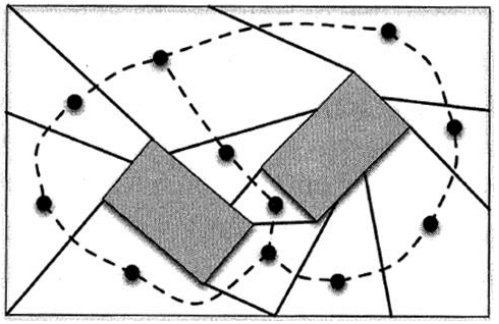
Discrétisation de l'espace

Représentation continue
(espace des configurations)



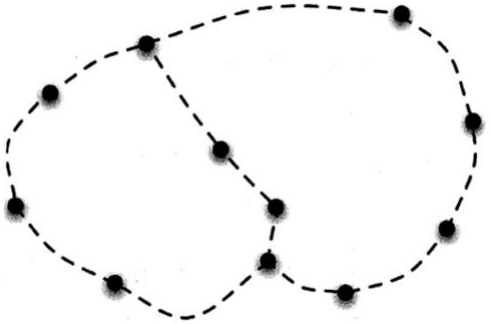
↓

Discrétisation
+ graphe



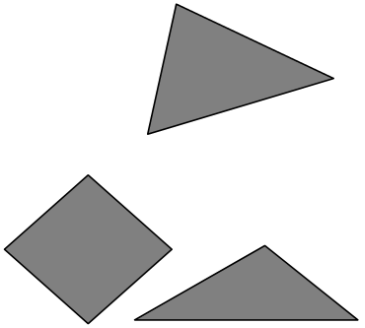
↓

Recherche dans graphe
(blind, best-first, A*)

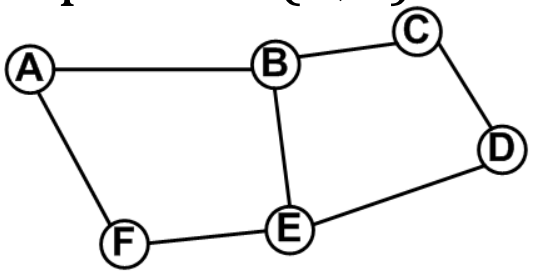


Graphe des visibilités

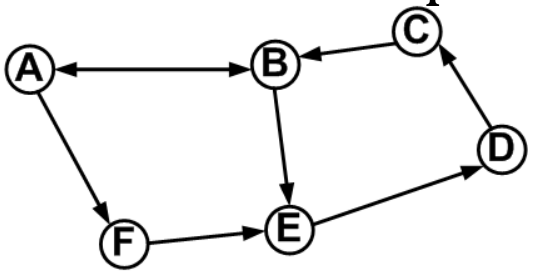
- Vous avez une carte géométrique connue
- Obstacles sont des **polygones**



- Utilise graphe G = (E,V) non-orienté comme représentation



graphe non-orienté



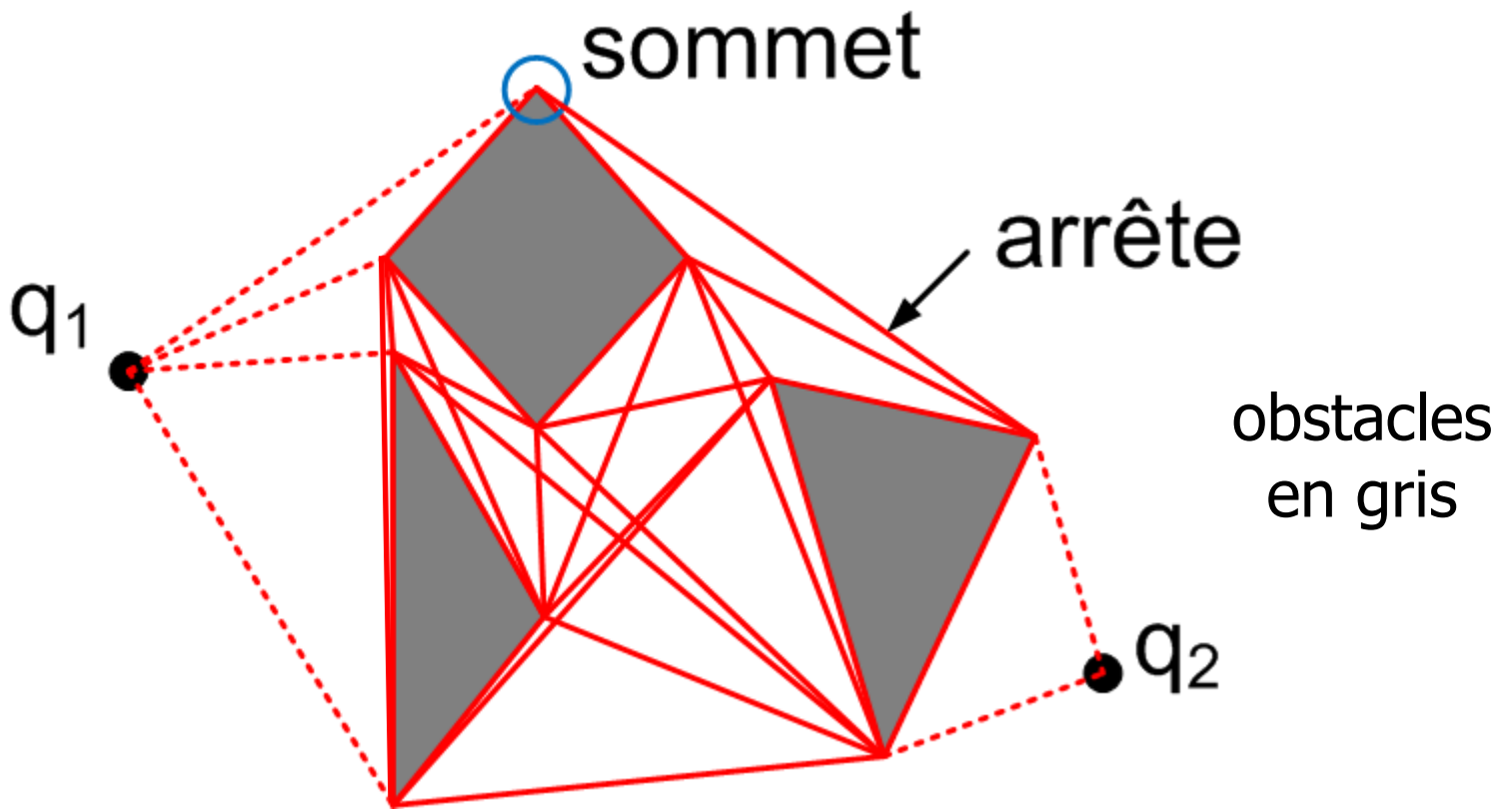
graphe orienté

- Algorithme complet*
- Approprié pour monde 2D
- Approche par « carte routière » : *roadmap*

*complet : qui retourne une solution, si elle existe

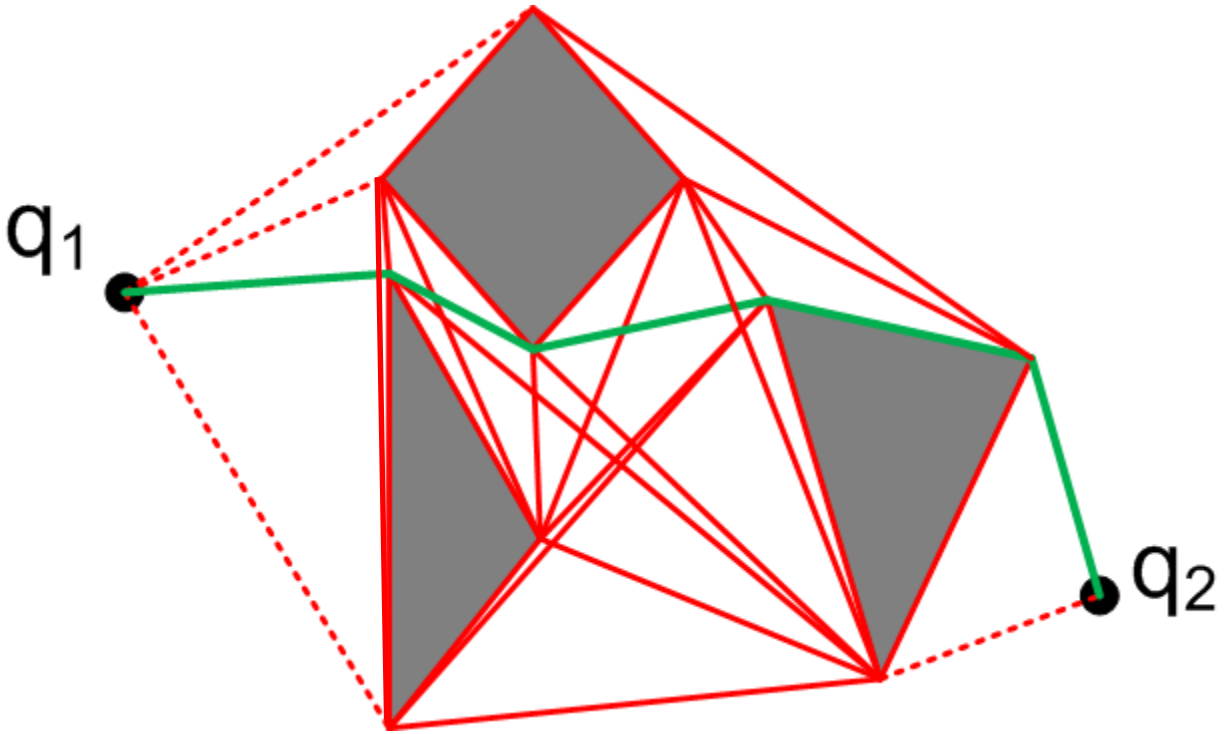
Discrétisation : graphe des visibilités

- Les sommets V des obstacles qui sont mutuellement visibles sont connecté ensemble par une arrête E.



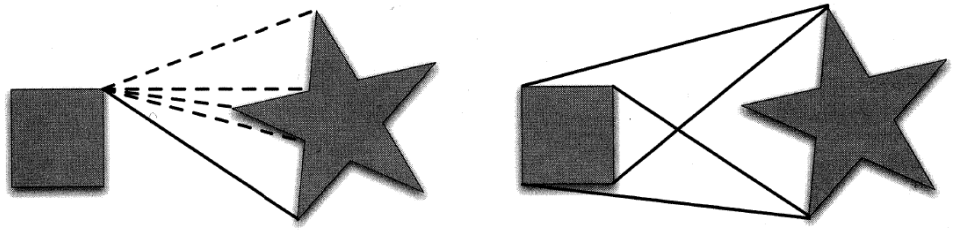
Planification : graphe des visibilités

- Planification : recherche le chemin le plus court dans graphe G entre q_1 et q_2.
- Algorithme de *Dijkstra* : complexité $O(|E| + |V|log|V|)$



Graphe des visibilité réduites

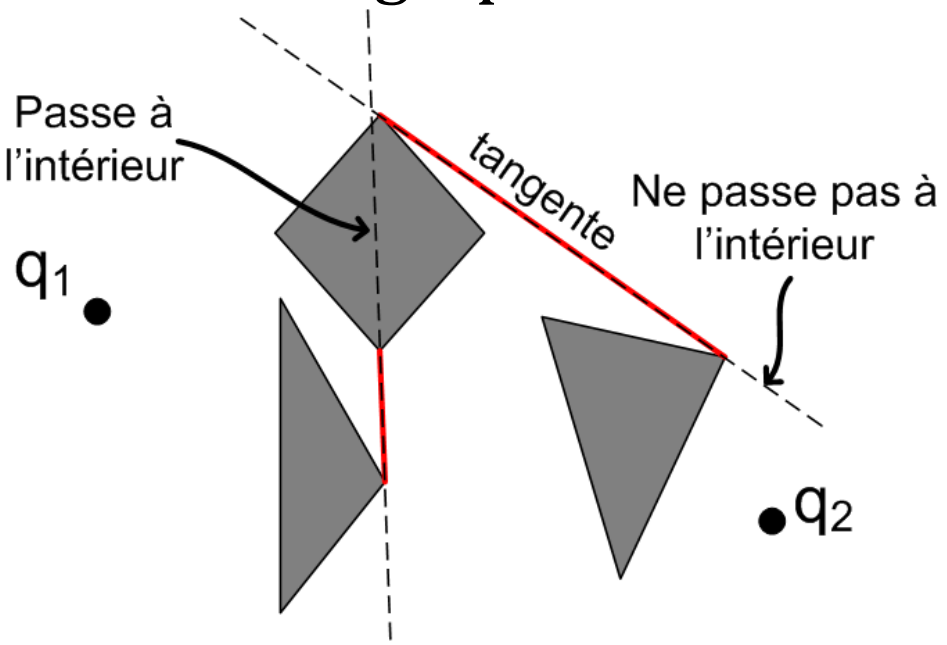
- Une ligne passant par deux sommets $V_1 \in \mathbf{A}$ et $V_2 \in \mathbf{B}$ est dite tangente si elle ne passe pas à l'intérieur d'un des deux obstacles **A** ou **B**.
- Vient réduire le nombre d'arrête E dans graphe G.



(a) Co-tangent lines from one vertex

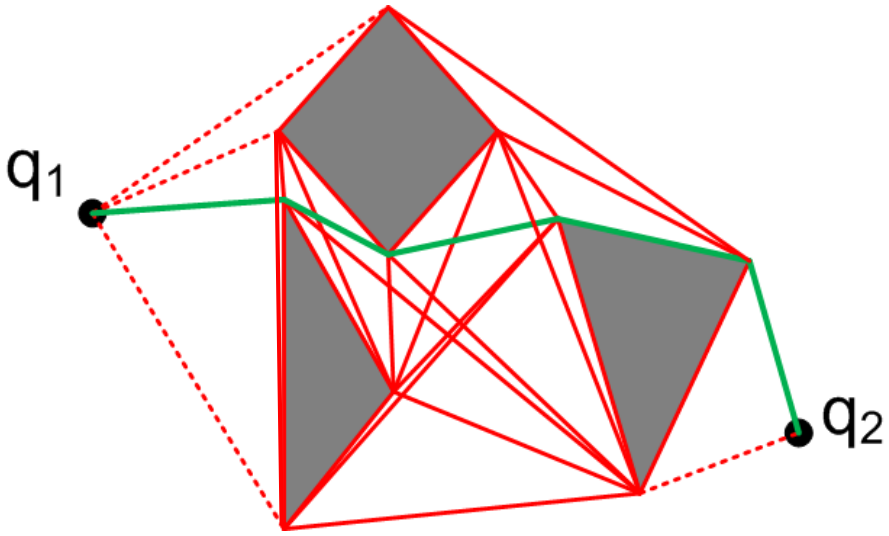
Co-tangent lines between two objects

Au maximum, 4 tangentes par paire d'obstacles

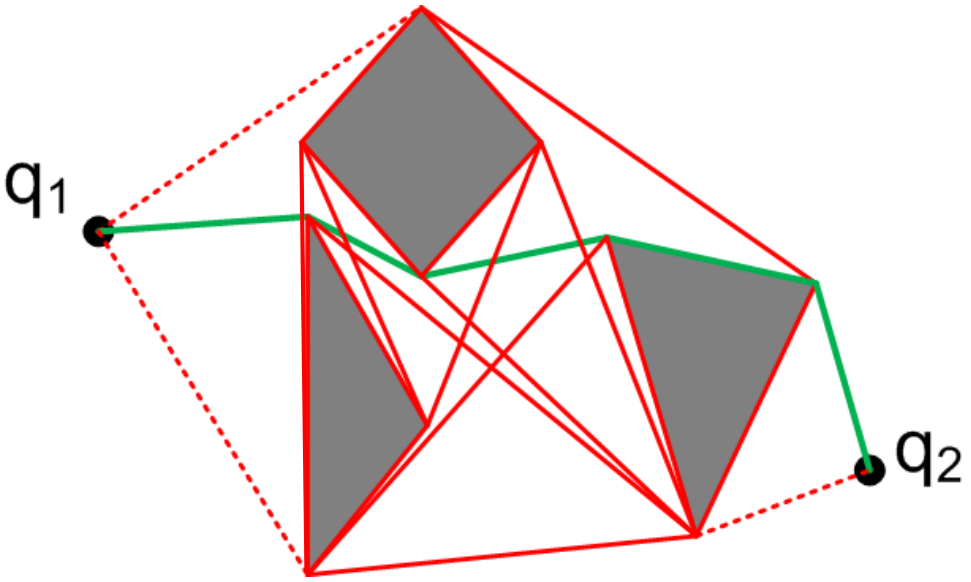


Graphe des visibilité réduites

- Graphe plus simple = recherche plus rapide
- Donne toujours la solution la plus courte



graphe des visibilités



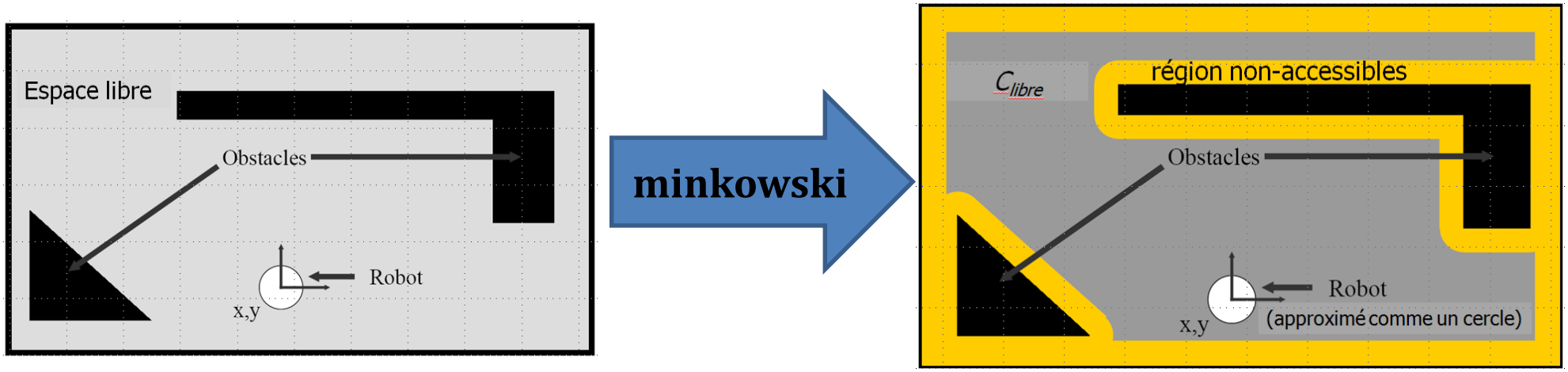
graphe des visibilités réduites

Graphe des visibilités

- Produit nécessairement la trajectoire la plus courte entre q_1 et q_2…
- …mais avec possiblement des trajectoires semi-libres…
- …et pour un robot de la taille d'un point.

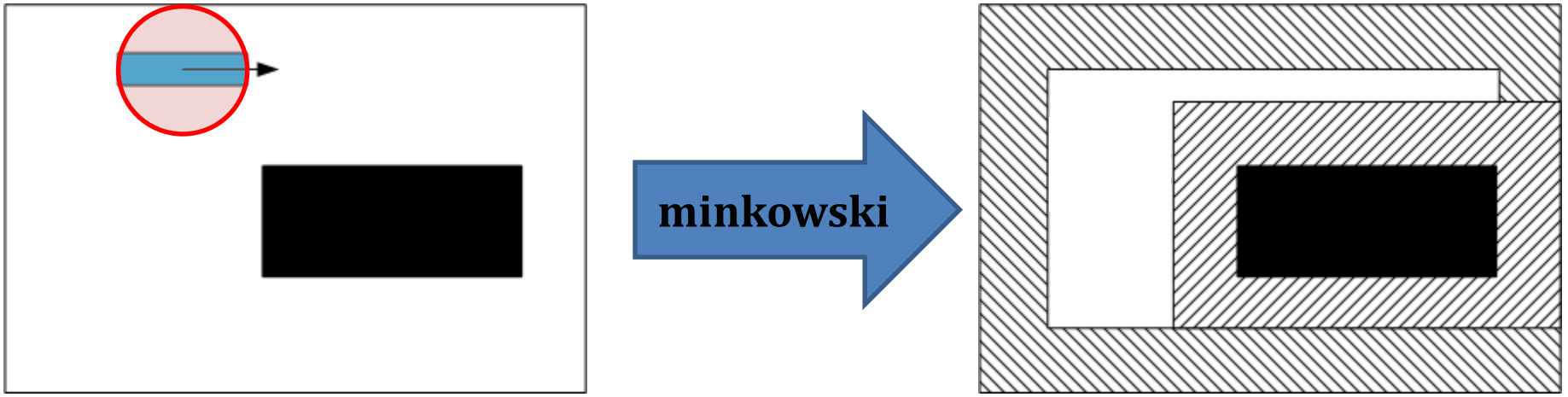
Somme de Minkowski

- Somme de Minkowski :
  - « gonfle » les obstacles pour tenir compte taille robot
  - robot redevient un point
- Algo <u>complet</u> seulement si le robot est circulaire

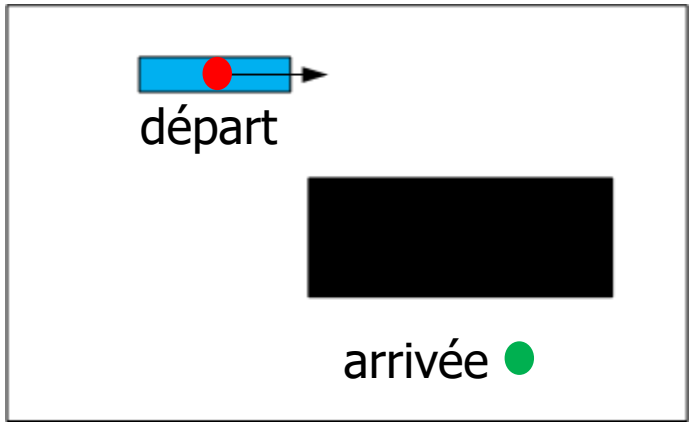


Somme de Minkowski

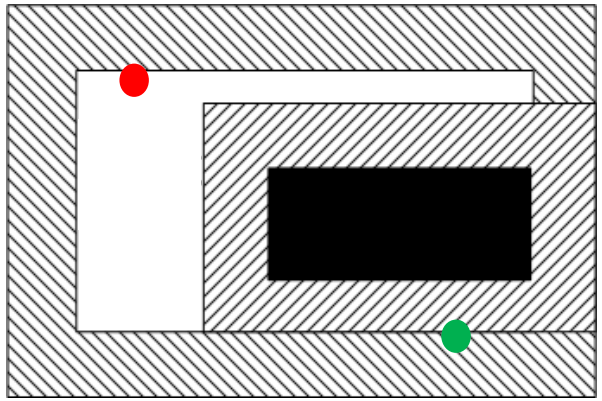
- Aucune garantie d'être complet si non-circulaire



Somme de Minkowski



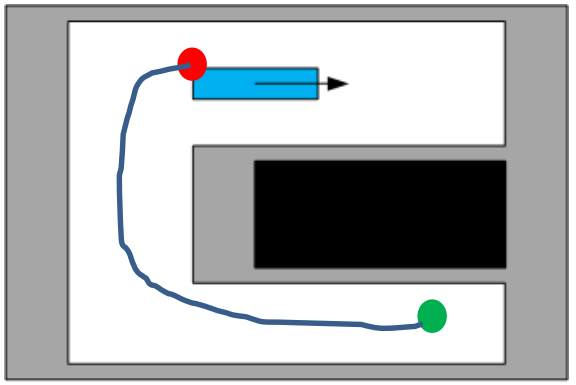
NON-COMPLET!

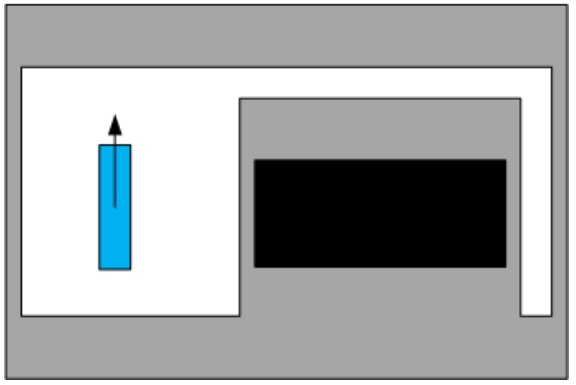


Minkowski : pas de solution

versus

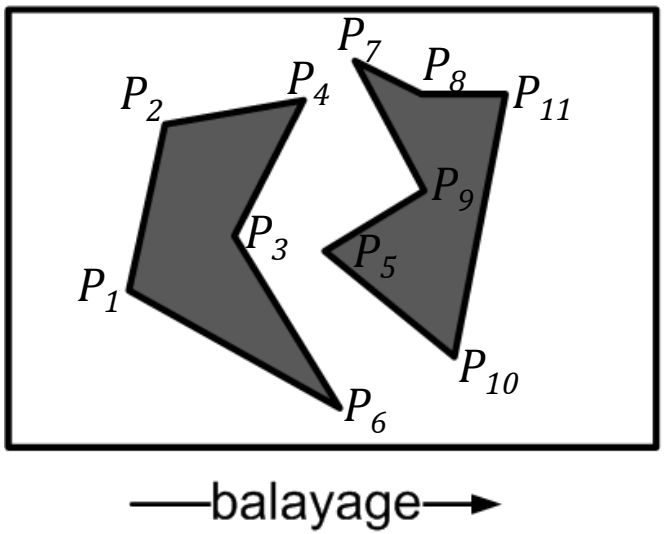
Espace de configuration en 3 D: solution possible



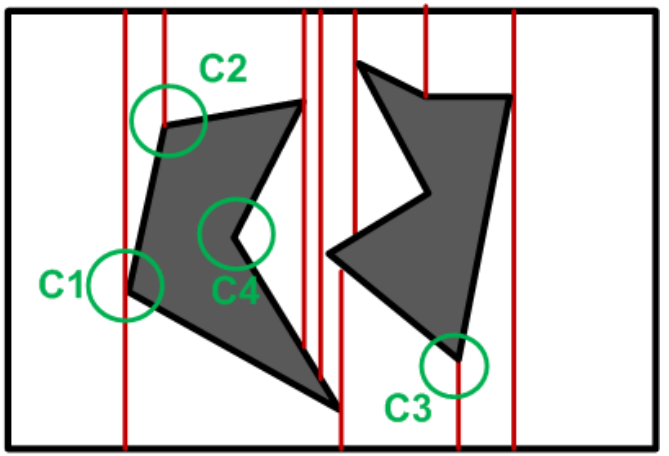


Décomposition cellulaire verticale

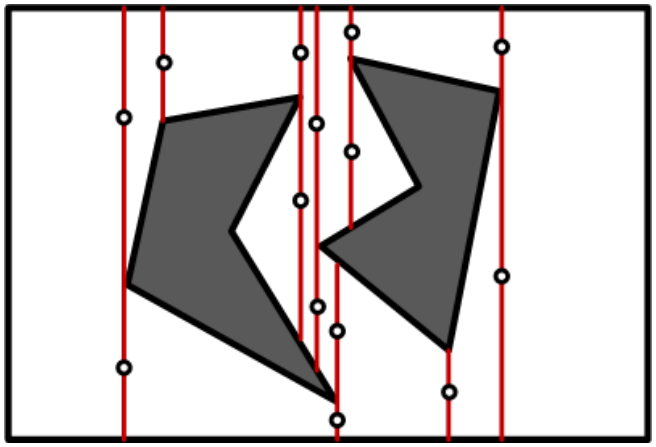
- Décomposer une carte 2D en trapèzes et triangles (éléments convexes),
- Pour chaque sommet P_i des obstacles:
  - étend ligne verticale en haut et en bas, jusqu'à un obstacle
- 4 cas possibles:
  - **C1**:(haut,bas)
  - **C2**:(haut),
  - **C3**:(bas),
  - **C4**:(aucun)



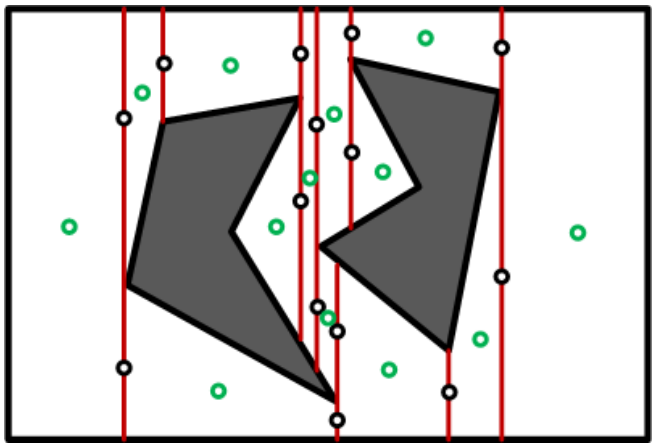
Étape 1



Décomposition cellulaire verticale

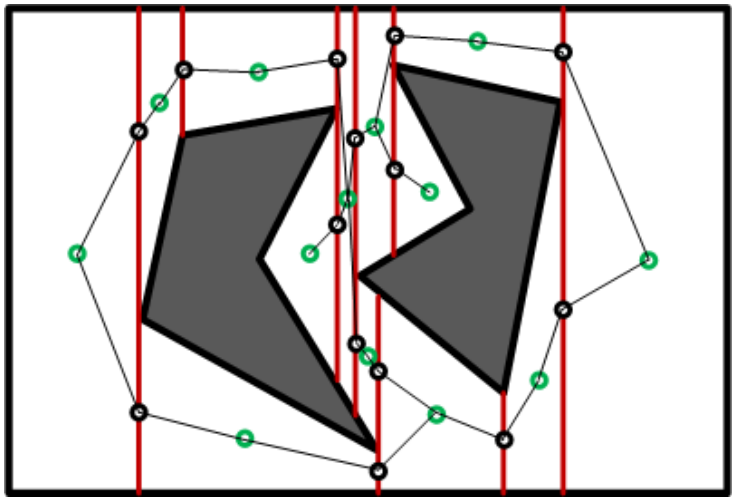


Étape 2:
place 1 point au milieu frontière entre cellules



Étape 3:
place 1 **point** au milieu de la cellules

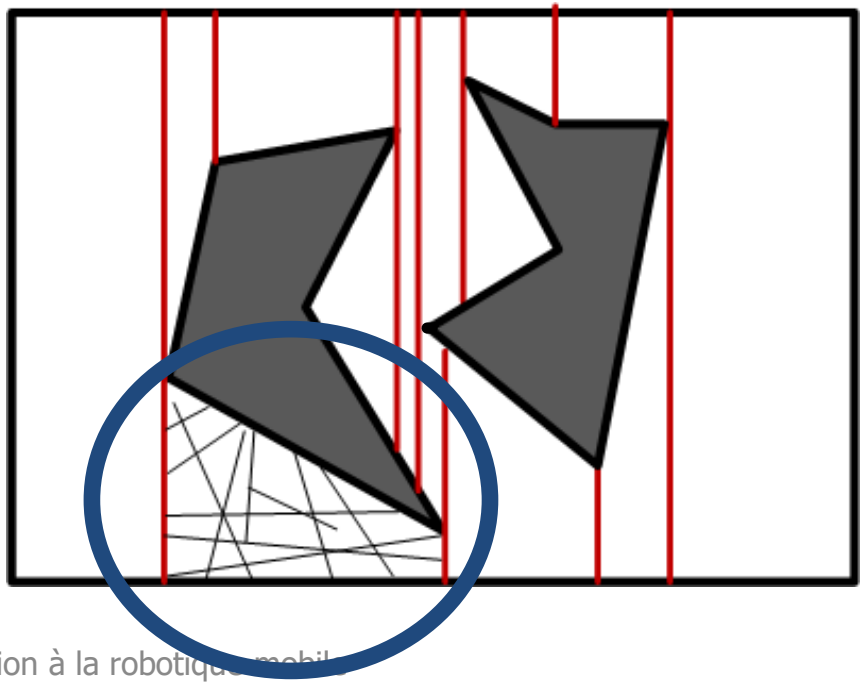
- Créer une carte routière



Étape 4:
relie les **centres** aux points frontières

Décomposition cellulaire verticale

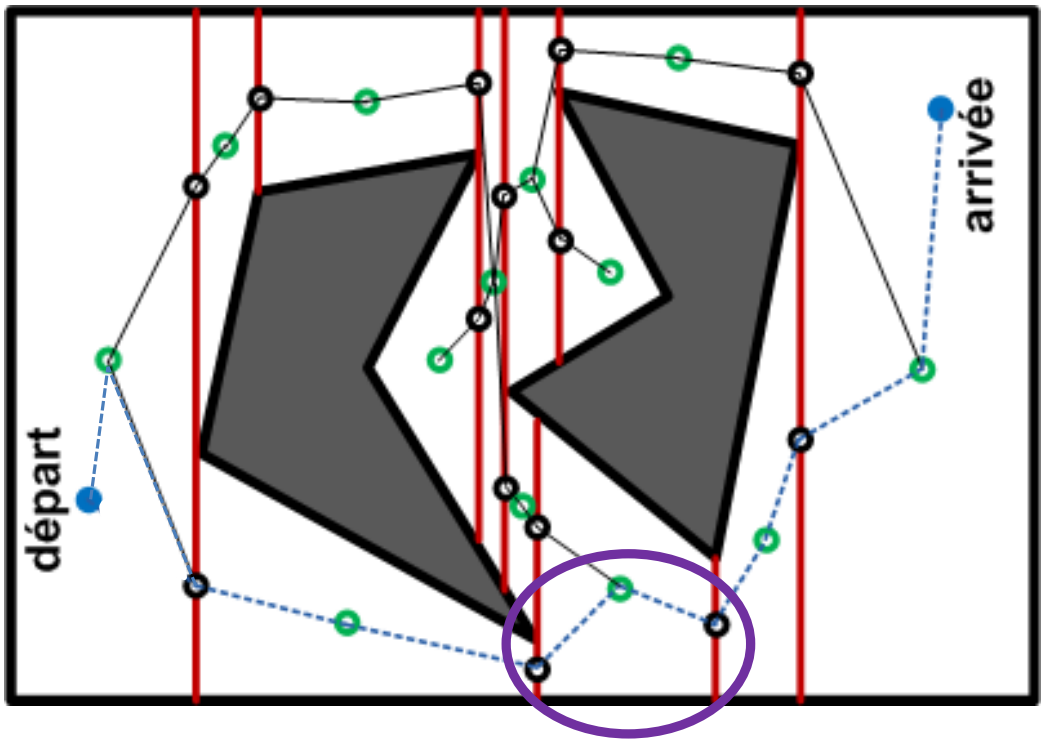
- Chaque cellule est convexe
  - ce qui veut dire que dans une cellule, on peut rejoindre n'importe quel autre point à l'intérieur, en ligne droite.



Décomposition cellulaire verticale

- Planification : recherche du chemin le plus court dans le graphe
- Donne des trajectoires qui sont en général loin des obstacles
- Création de la carte : complexité $O(n \log n)$

approche roadmap



pas optimal en distance

Champs de potentiels

- Imaginez que :
  - **robot** est une particule **+**
  - **obstacles** sont aussi chargés **+** (répulsion)
  - **but** est chargé **-** (attraction)

- Trajectoire est le chemin parcouru par la particule (robot) dans le champs de potentiel

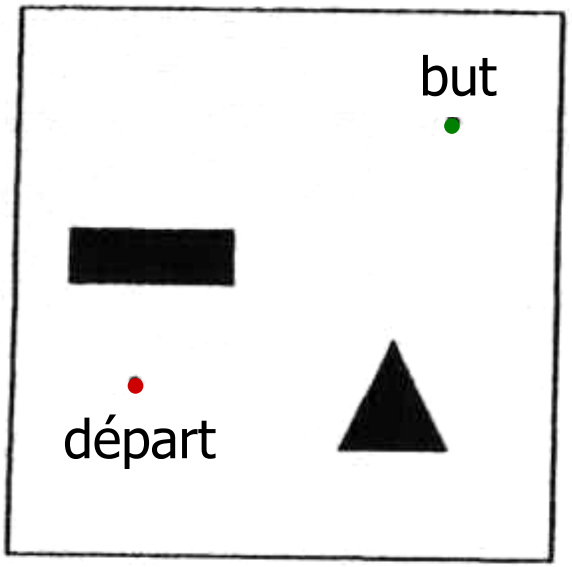
Champs de potentiels

Calcul des forces de répulsions et d'attraction

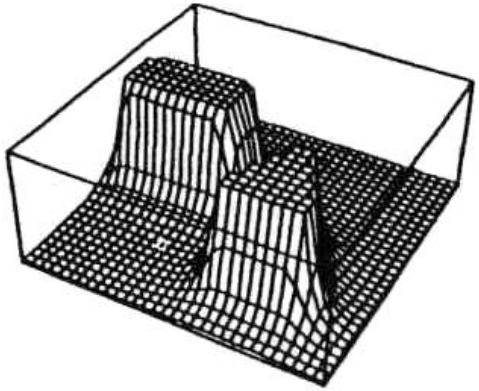
$$U(q) = U_{but}(q) + \sum U_{obstacles}(q)$$

$$F = -\nabla U(q) = \begin{bmatrix} \frac{\partial}{\partial x} U(q) \\ \frac{\partial}{\partial y} U(q) \end{bmatrix}$$

gradient ∇ indique la direction maximale de croissance d'un champ scalaire



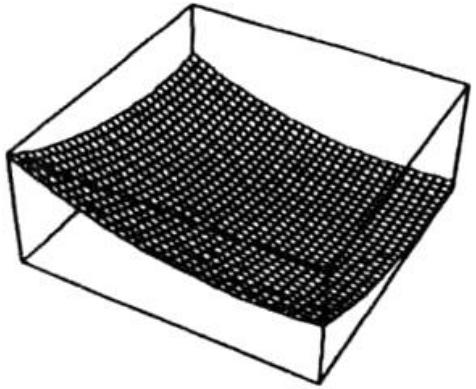
potentiel de répulsion



$\sum U_{obstacles}(q)$

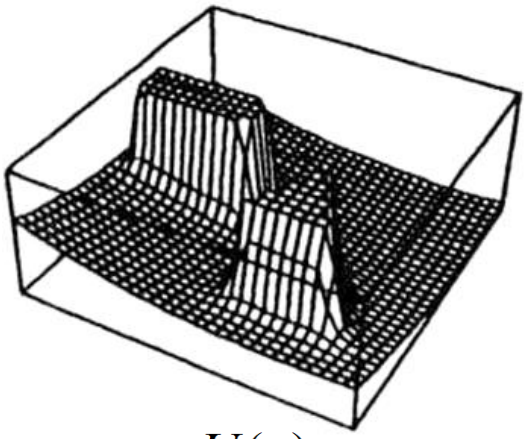
+

potentiel d'attraction



$U_{but}(q)$

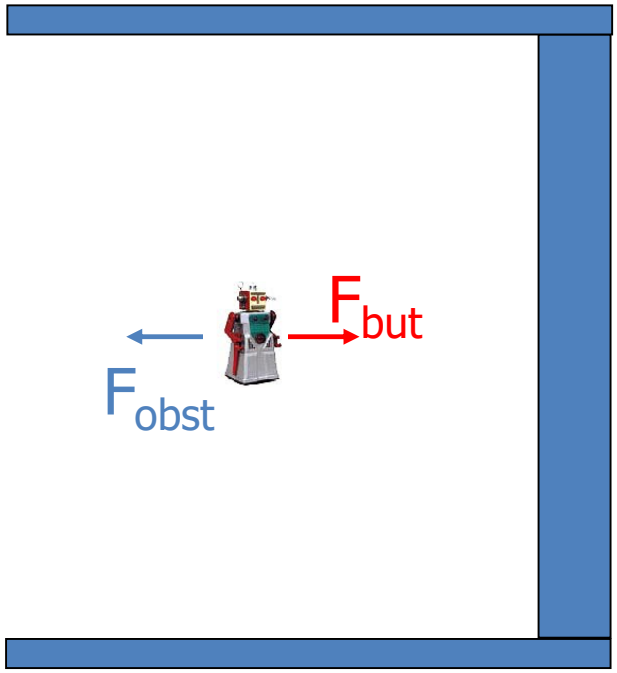
=

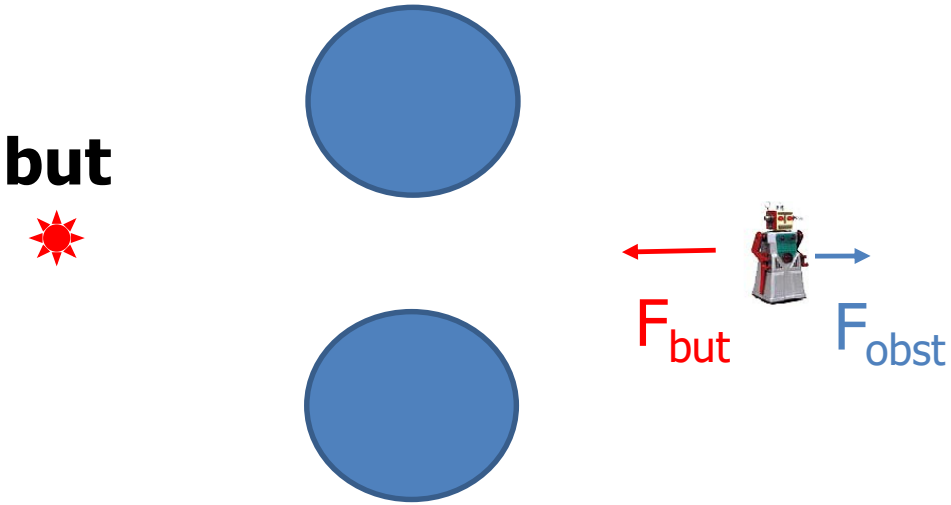


$U(q)$

Champs de potentiels

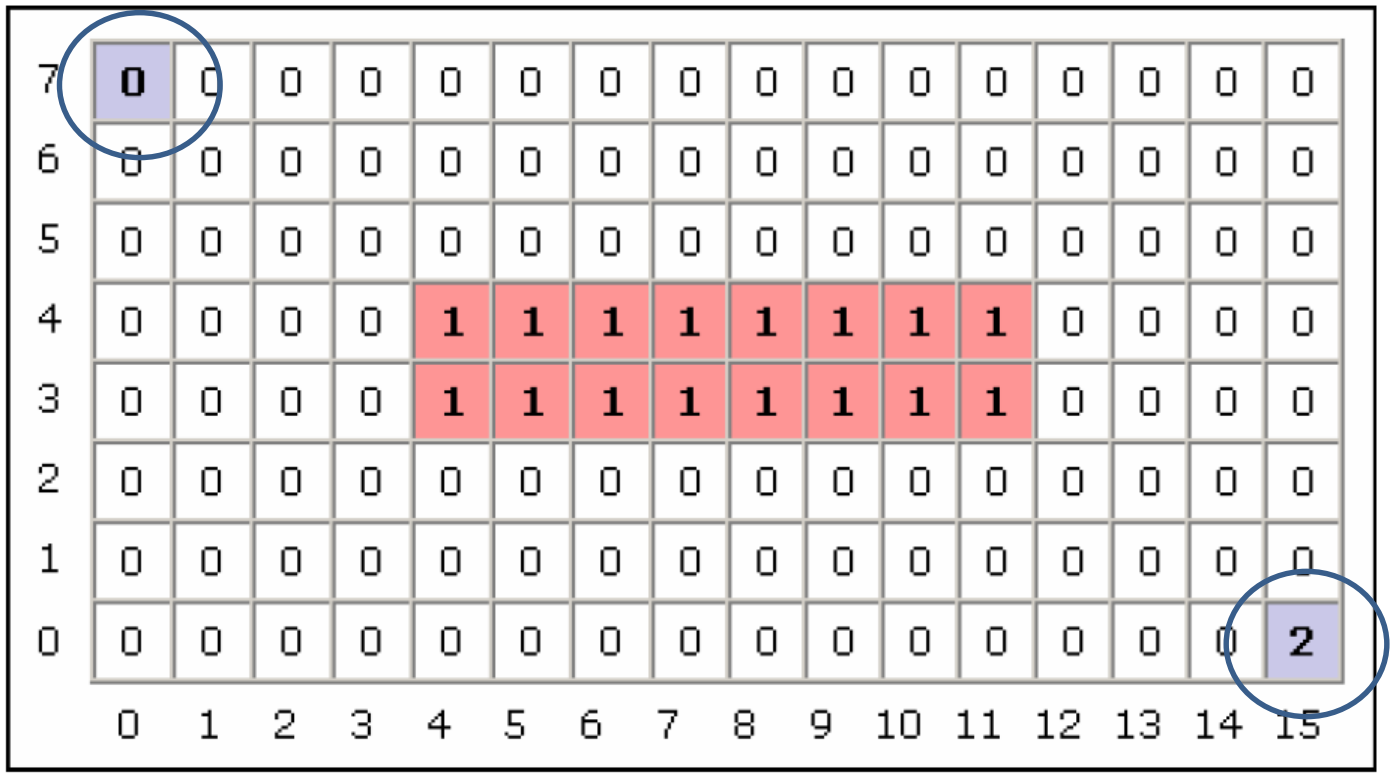
- Tombé en partie en désuétude parce que sensible aux minimum locaux.





Planification par front d'onde

- *Wavefront planner*
- **0**: non-visité 1: obstacle 2: indique le départ



Planification par front d'onde

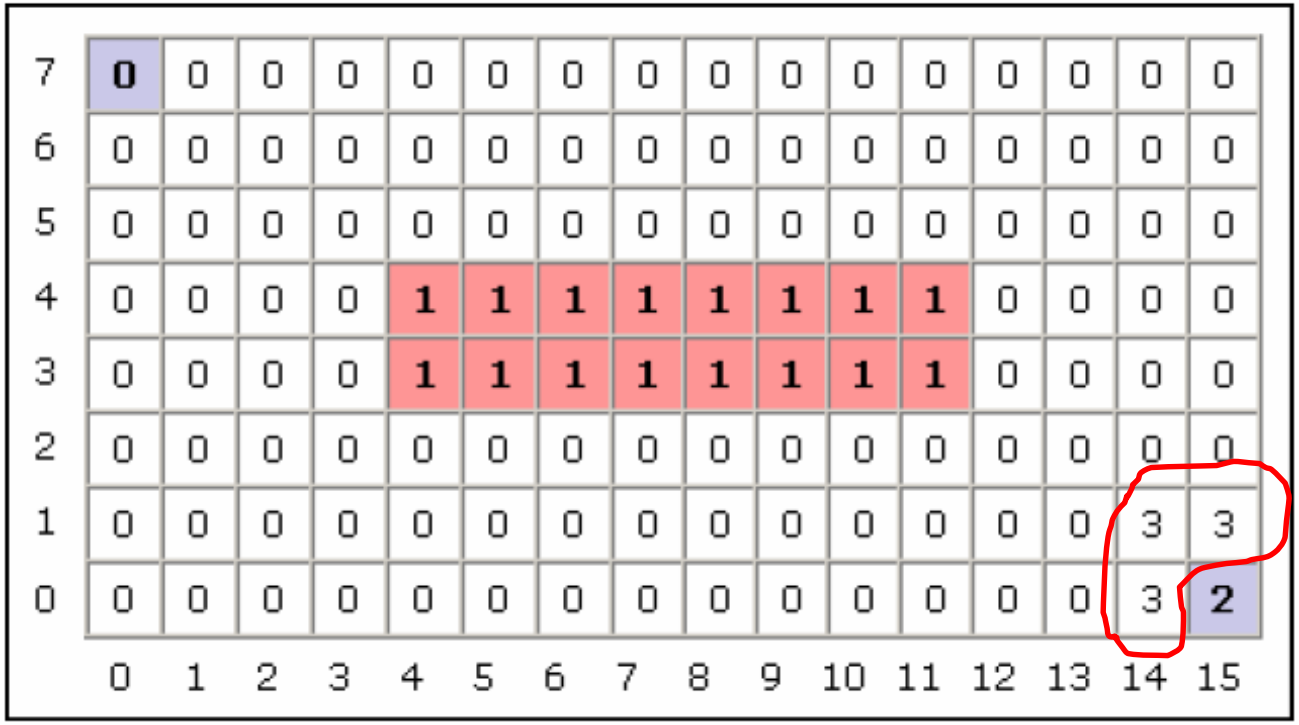
- *Wavefront planner*
- **0**: non-visité 1: obstacle 2: indique le départ

connectivité Von Neumann

	1	
4	*C*	2
	3	

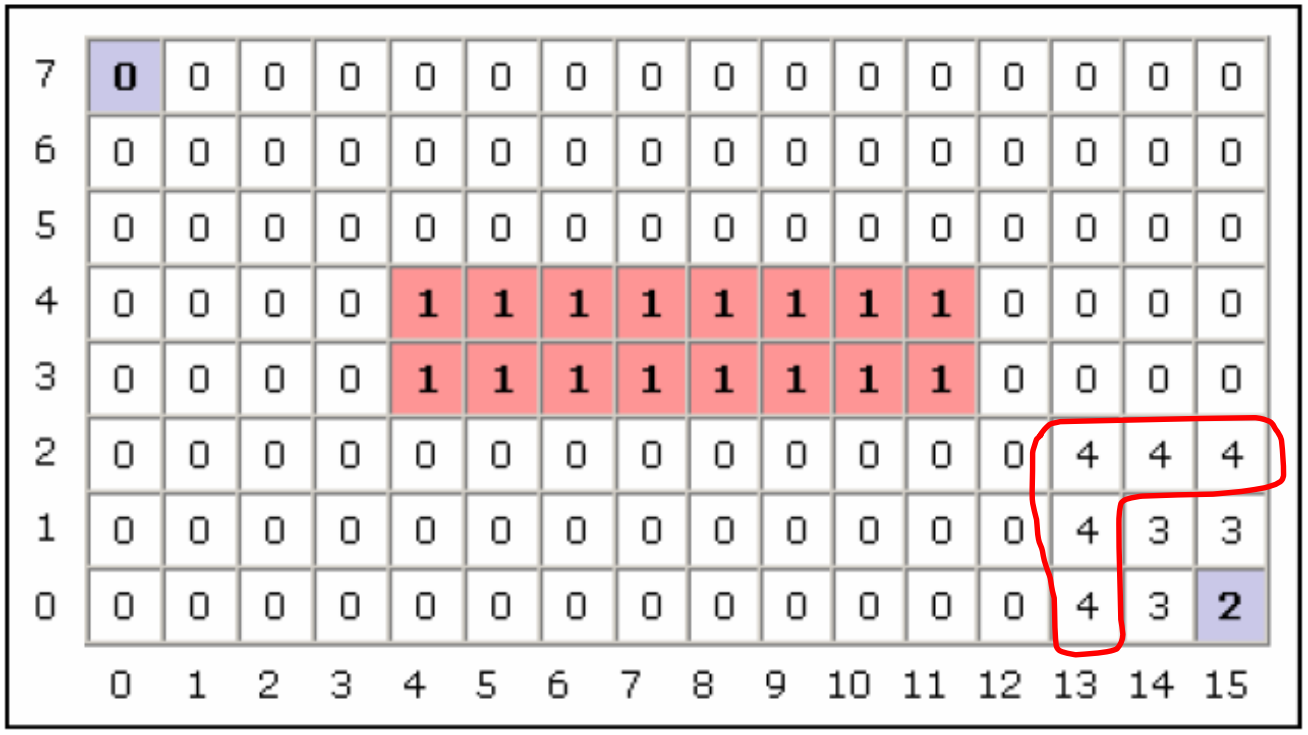
connectivité Moore

1	2	3
4	*C*	5
6	7	8



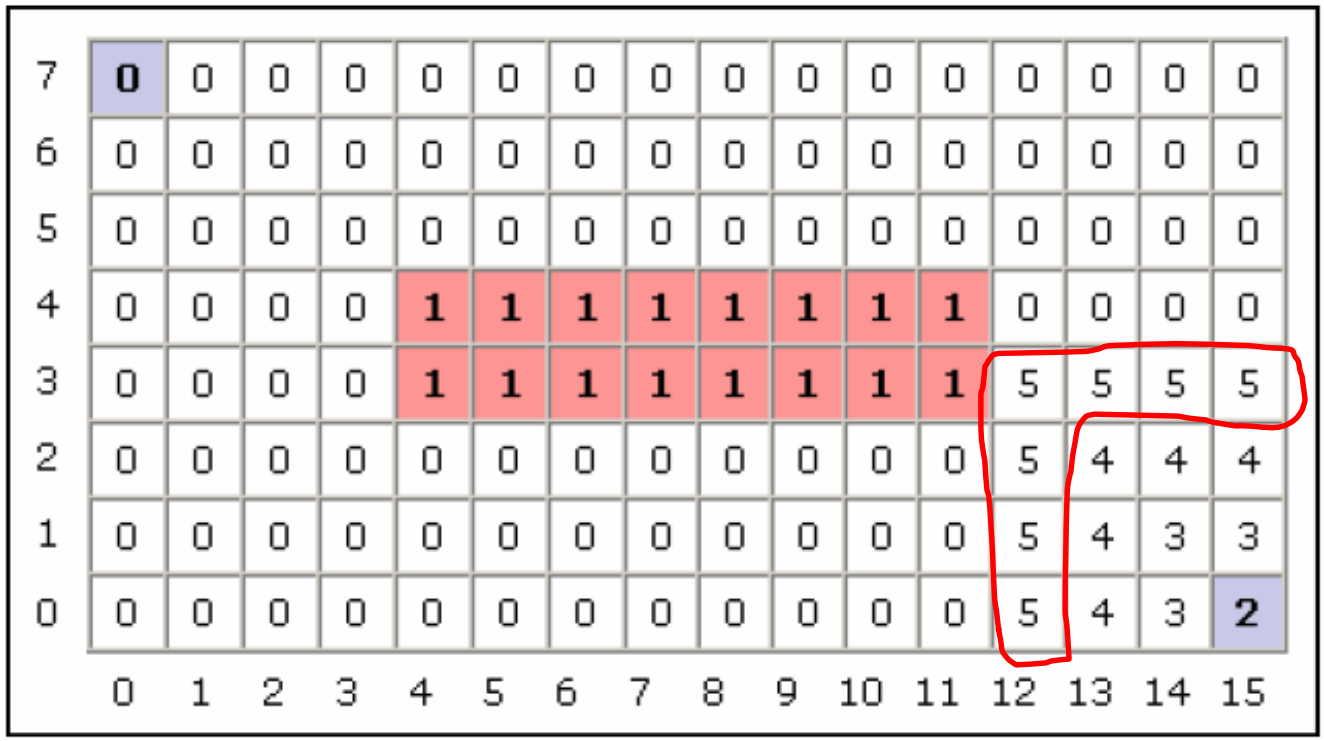
Planification par front d'onde

- *Wavefront planner*
- **0**: non-visité `1`: obstacle `2`: indique le départ



Planification par front d'onde

- *Wavefront planner*
- **0**: non-visité `1`: obstacle `2`: indique le départ



Planification par front d'onde

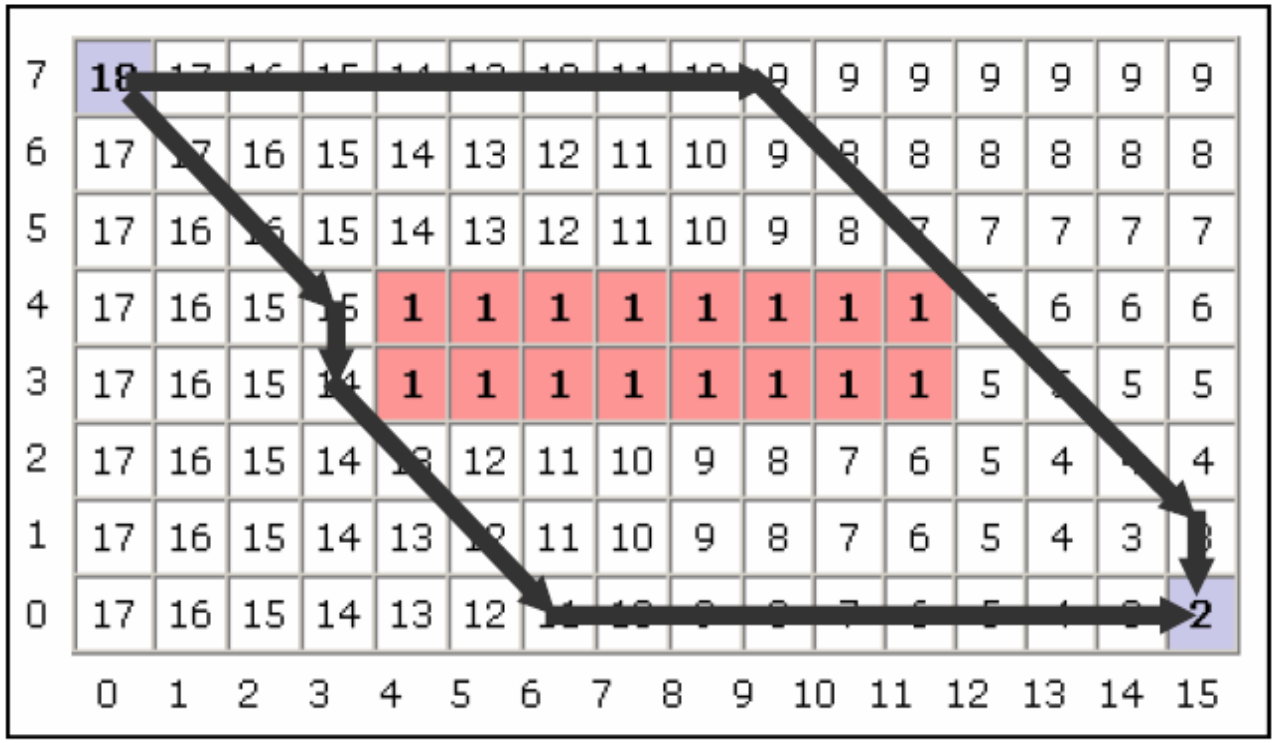
- Rempli jusqu'à plein
  - les 0 indiquent les endroits non-accessibles du point de départ.

7	**18**	17	16	15	14	13	12	11	10	9	9	9	9	9	9	9
6	17	17	16	15	14	13	12	11	10	9	8	8	8	8	8	8
5	17	16	16	15	14	13	12	11	10	9	8	7	7	7	7	7
4	17	16	15	15	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	6	6	6	6
3	17	16	15	14	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	5	5	5	5
2	17	16	15	14	13	12	11	10	9	8	7	6	5	4	4	4
1	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	3
0	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	**2**
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

UNIVERSITÉ LAVAL

Planification par front d'onde

- La trajectoire consiste **à partir du but** de choisir de façon vorace les cases les plus petites.



2 des nombreuses trajectoires possibles

Planification par front d'onde

- Si vous faites le choix vorace des nœuds **à partir du départ** ☹, vous n'aurez pas nécessairement un chemin le plus court