La méthode Q-Learning

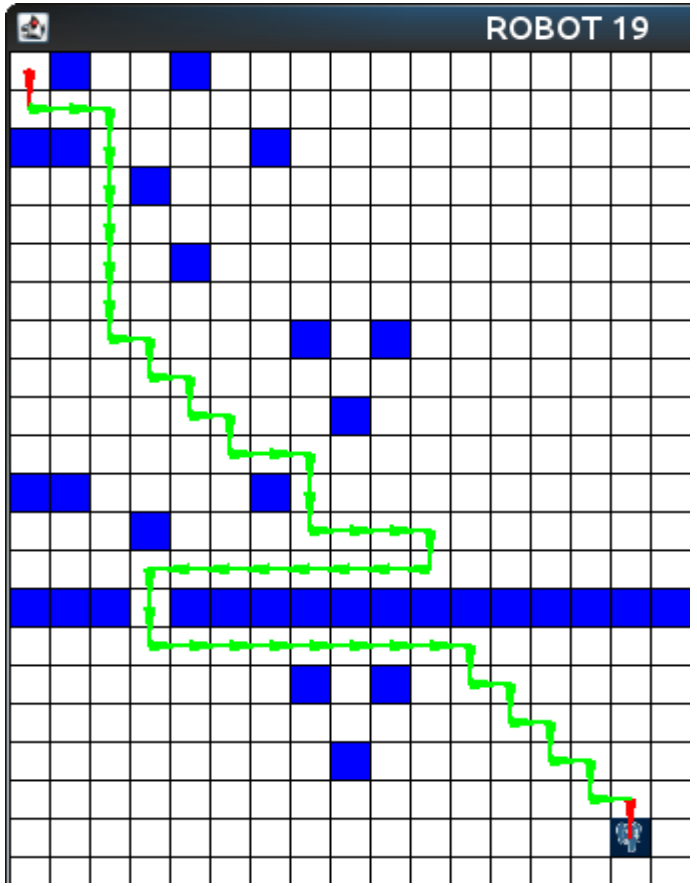
Plan

- Introduction
- Méthodes d’optimisation
- Apprentissage par renforcement
- La méthodes Qlearning
  - Exemple d’application
  - Implémentation
  - Mini Projet à faire

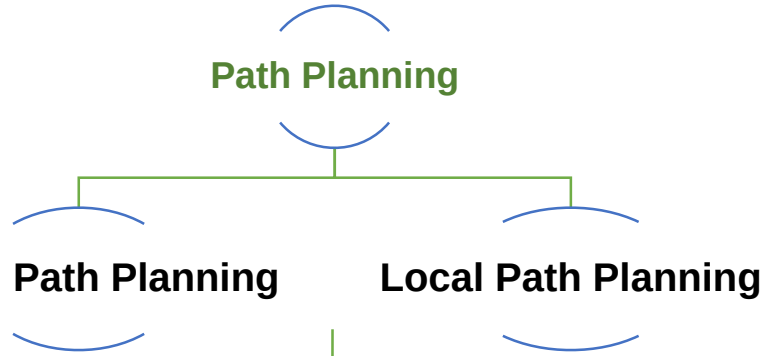
Introduction

- Pour un robot **A** évoluant dans un environnement **W** donné, le problème général de planification consiste à déterminer pour A un mouvement lui permettant de se déplacer entre deux configurations données tout en respectant un certain nombre de contraintes et de critères.

- Les critères à satisfaire pendant la résolution du problème de planification concernent le fait qu'une solution doit optimiser une fonction de coût exprimée en terme de la distance parcourue par le robot entre les deux configurations extrémités, de la durée ou de l'énergie nécessaires à l'exécution de son mouvement.



Introduction



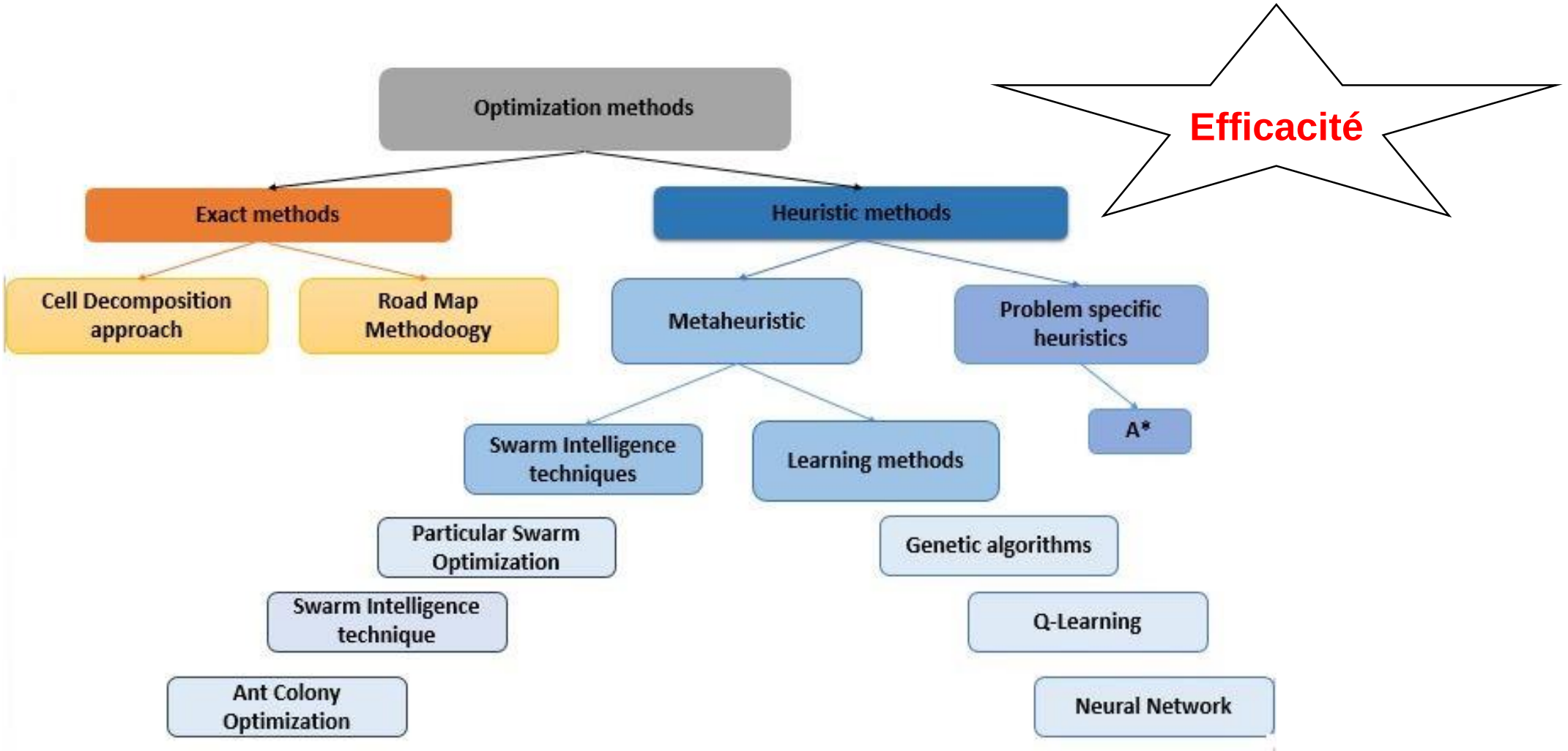
Environnement statique : Les obstacles sont connus.

Le robot connais l'emplacement de tous les Obstacles avant le début du navigation.

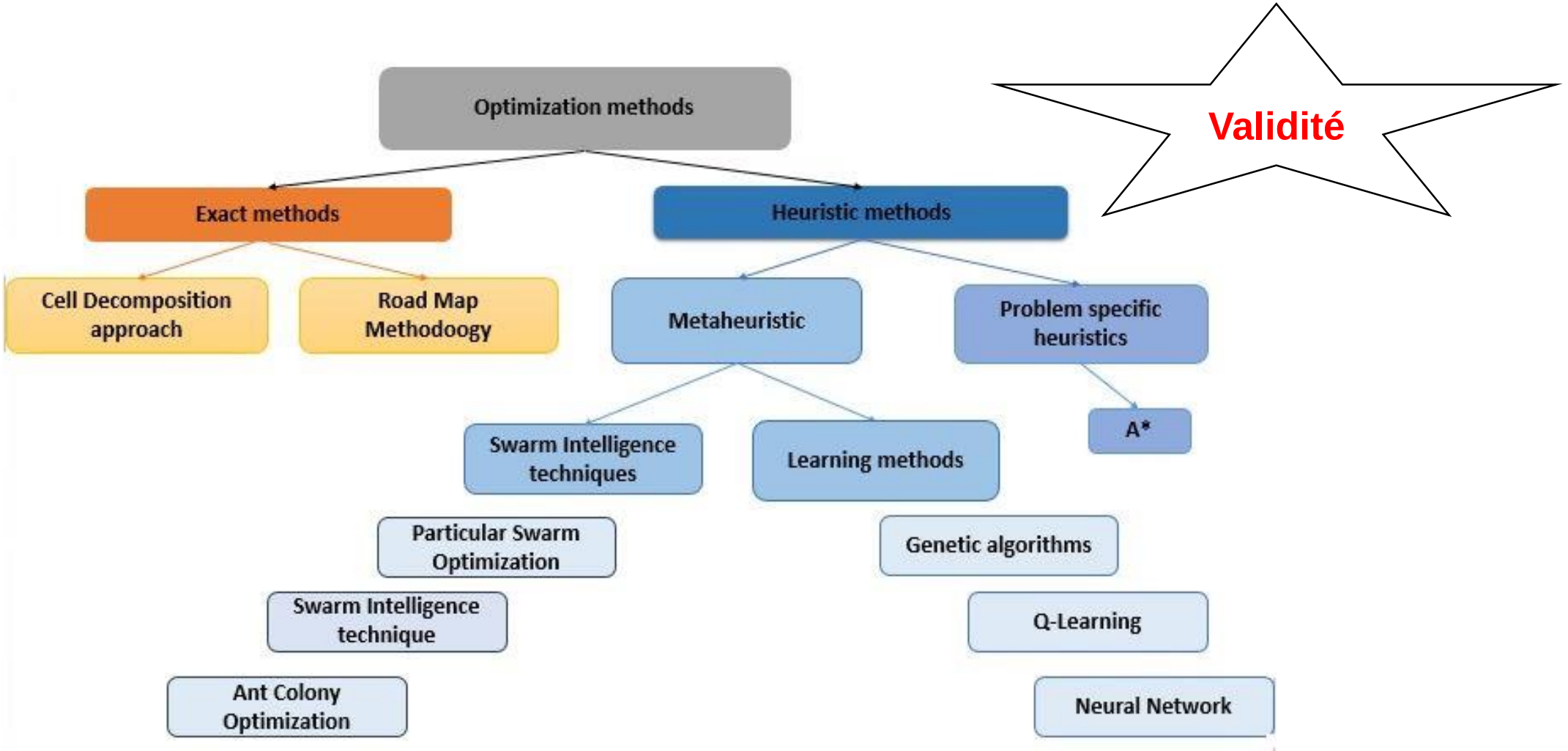
Environnement dynamique : Les obstacles sont inconnus

Le robot découvre l'environnement au fur et mesure de son mouvement en utilisant des Sensors pour localiser l'emplacement, la forme et la taille des obstacles

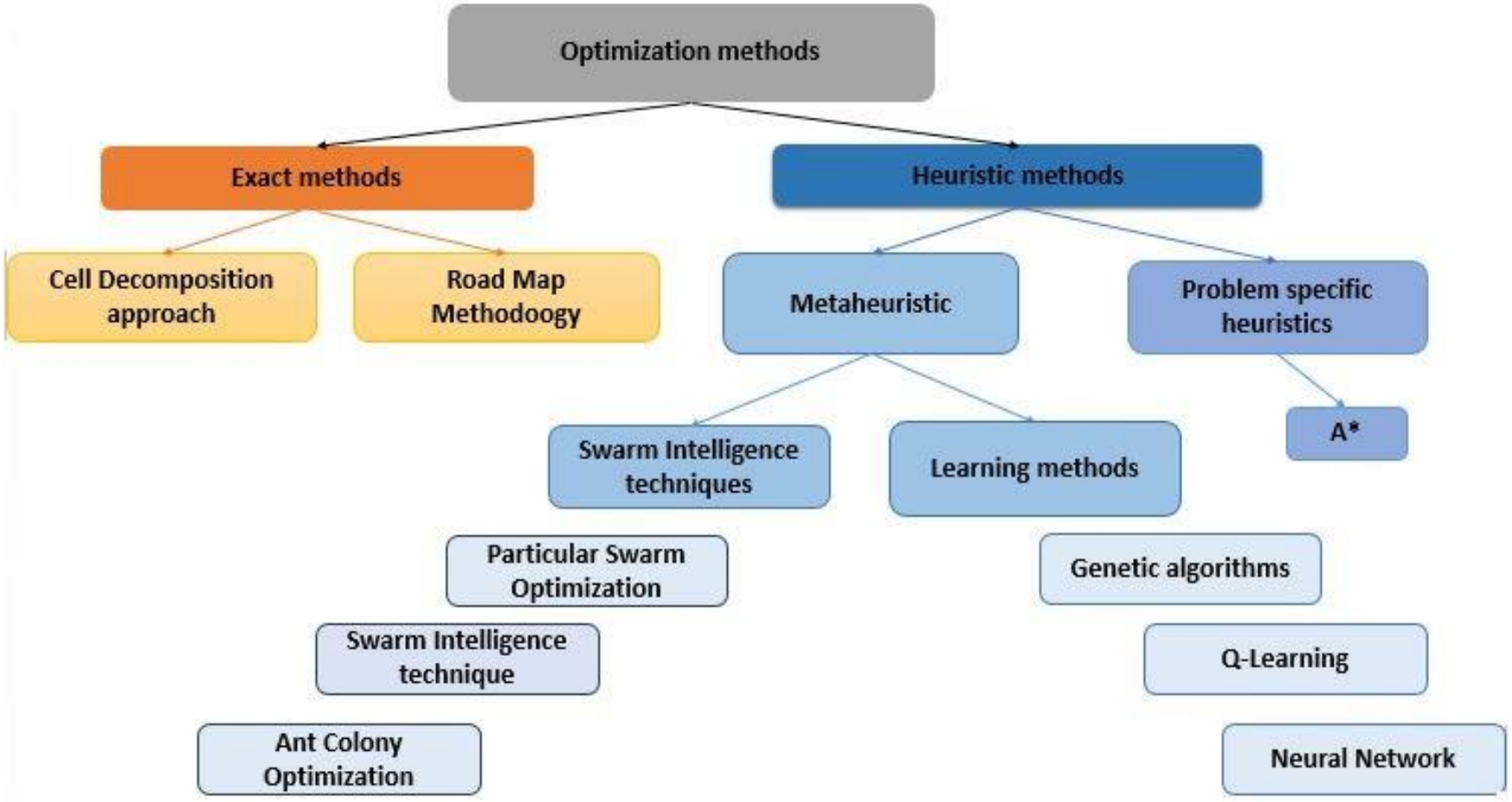
Méthodes d’optimisation



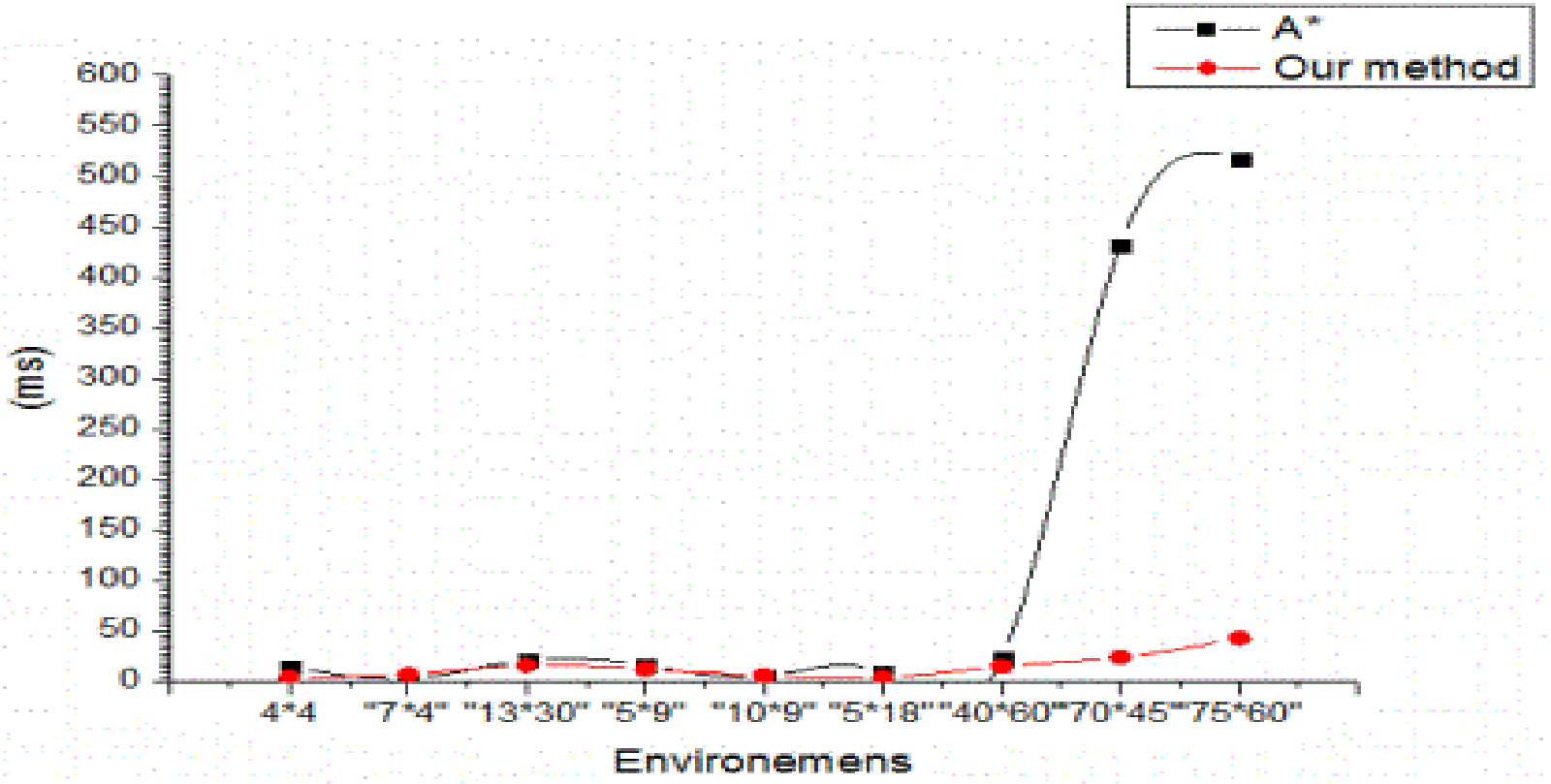
Méthodes d’optimisation



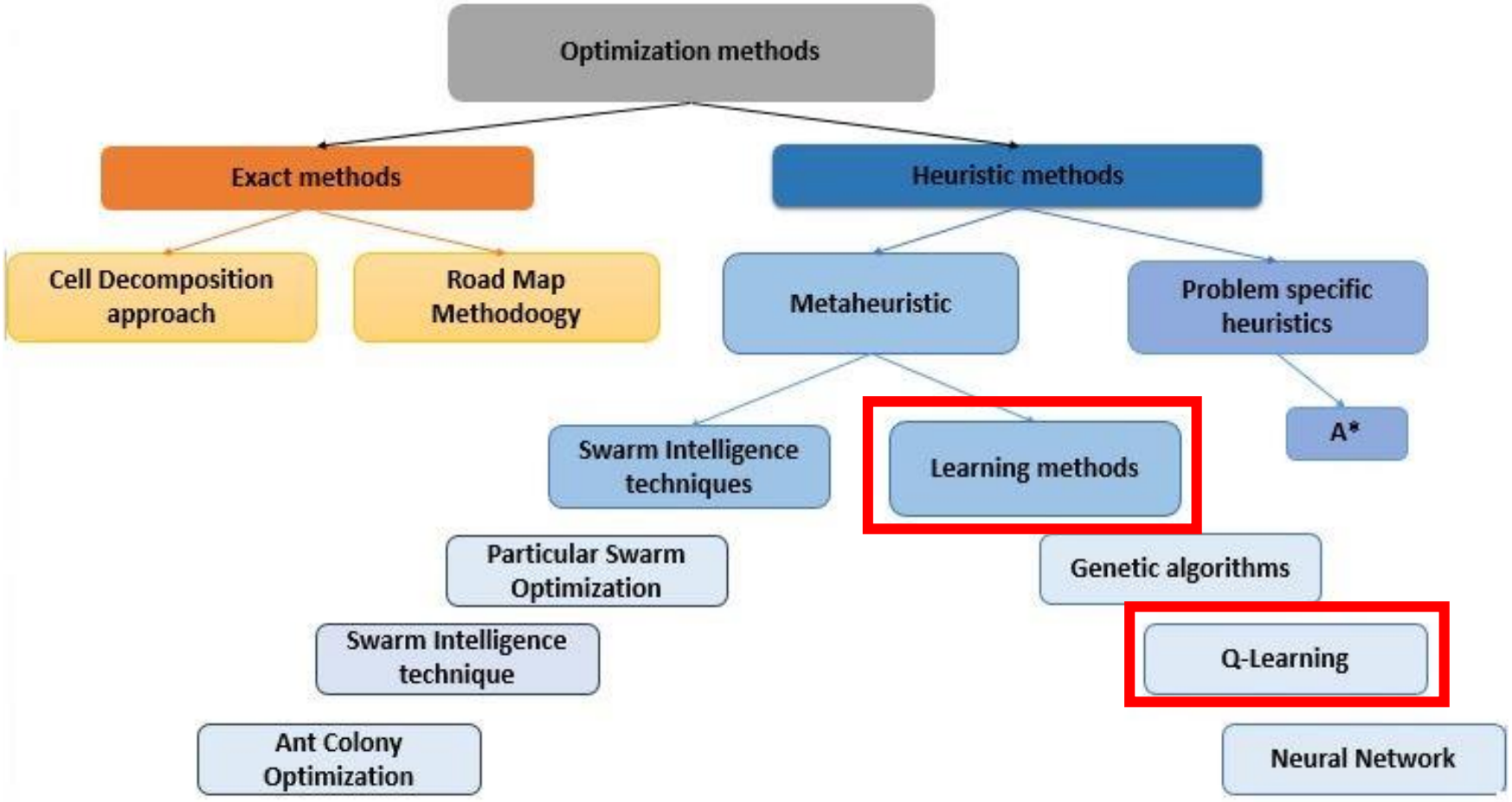
Méthodes d’optimisation



Méthodes d’optimisation



Méthodes d'optimisation



Apprentissage par renforcement

- Un agent **apprend** s'il améliore sa performance sur des tâches futures avec l'expérience,

Pourquoi programmer des agents qui apprennent ?

Apprentissage par renforcement

- Il existe plusieurs sortes d'apprentissage :

  - L'apprentissage supervisé :
    ex.: reconnaître les âges des personnes à l'aide des exemples de photos.

  - L'apprentissage non supervisé:
    ex.: identifier différents thèmes d'articles de journaux en regroupant les articles similaires (« clustering »)

  - L'apprentissage par renforcement:
    ex.: Robot qui apprend à naviguer dans un environnement

Apprentissage par renforcement

Motivation :

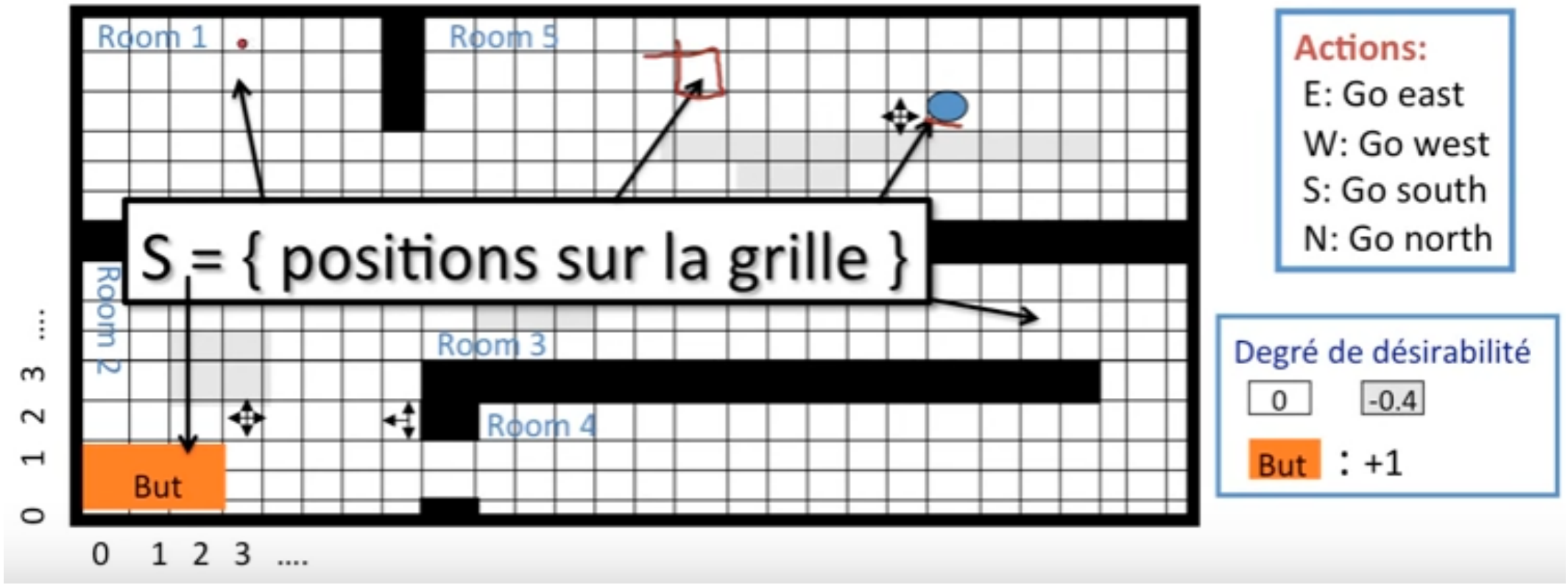
Pour obtenir un agent intelligent qui joue bien aux échecs, il faudrait amasser des paires (état du jeu, mouvement à jouer) d'un joueur expert

Amasser de telles données peut être fastidieux ou trop coûteux.

On préférerait que l'agent apprenne seulement à partir du résultat de parties qu'il joue.

- Si l'agent a gagné, c'est que son plan (sa politique) de jeu était bon.
- Si l'agent perd, c'est qu'il y a une faiblesse derrière sa façon de jouer.

Apprentissage par renforcement

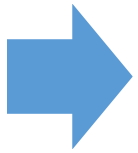


Apprentissage par renforcement

Cause à effet

- Le terme «cause à effet» pour l'apprentissage par renforcement peut être caractérisé par les étapes suivantes :

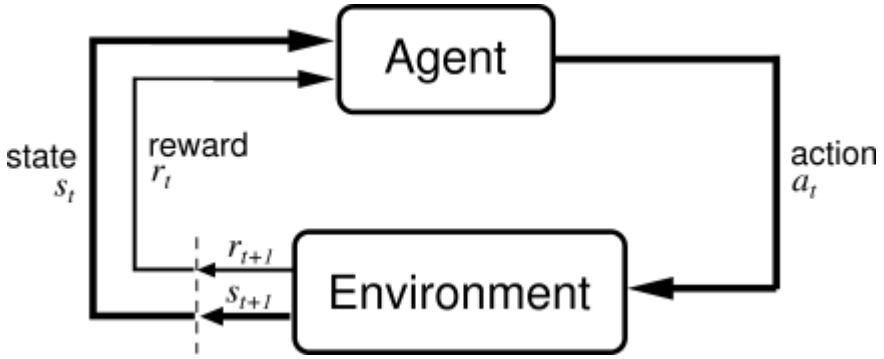
1. L'agent observe un état d'entrée.
2. Une action est déterminée par une fonction de prise de décision (politique).
3. L'action est effectuée.
4. L'agent reçoit une résultât en fonction de son environnement.
5. Informations sur le résultât donnée pour cette état ou action est enregistrée.



En effectuant des actions, on observe les récompenses qui en résultent, afin de déterminer la meilleure action pour un état donné.

Apprentissage par renforcement

- L'**apprentissage par renforcement** s'intéresse au cas où l'agent doit apprendre à agir seulement **à partir des récompenses** ou **renforcements**



- Données du problème d'apprentissage:
  - L'agent **agit** sur son environnement
  - Reçoit une retro-action sous-forme de **récompense (renforcement)**
  - Son **but** est de maximiser la somme des recompenses espérés

- Objectif: **Apprendre** à maximiser somme des recompenses

La méthode Qlearning

Qlearning : Principe

- L'agent ne connait pas les états ou se trouve les récompenses, ne connait pas a priori l'état d'arrivée d'une action.
- Il commence donc par choisir des actions aléatoirement, il explore.
- Au bout d'un certain temps ou lorsqu'il a atteint un état but, le système reprend une recherche de solution à partir de l'état initial.

La méthode Qlearning

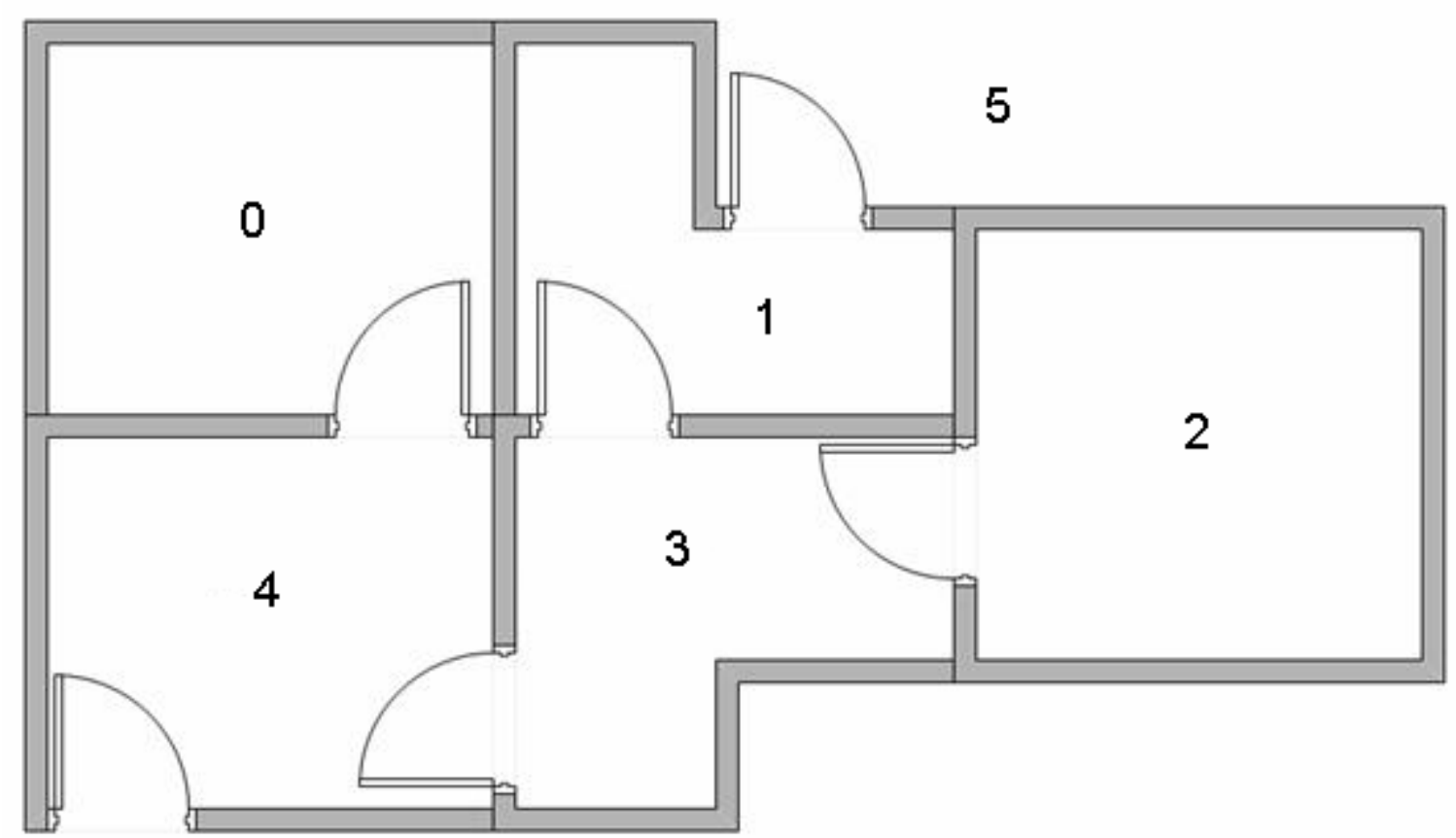
Qlearning : Principe

- La caractéristique distinctive de Q-Learning est sa capacité à choisir entre des récompenses immédiates et des récompenses retardées.

- A chaque étape du temps, un agent observe un état S, puis choisit et applique une action a. Alors que l'agent passe à l'état s + 1, l'agent reçoit une récompense R(s,a).

- Le but de l'apprentissage est de trouver l'ordre séquentiel des actions qui maximise la somme des récompense future, conduisant ainsi au chemin le plus court du début à la fin.

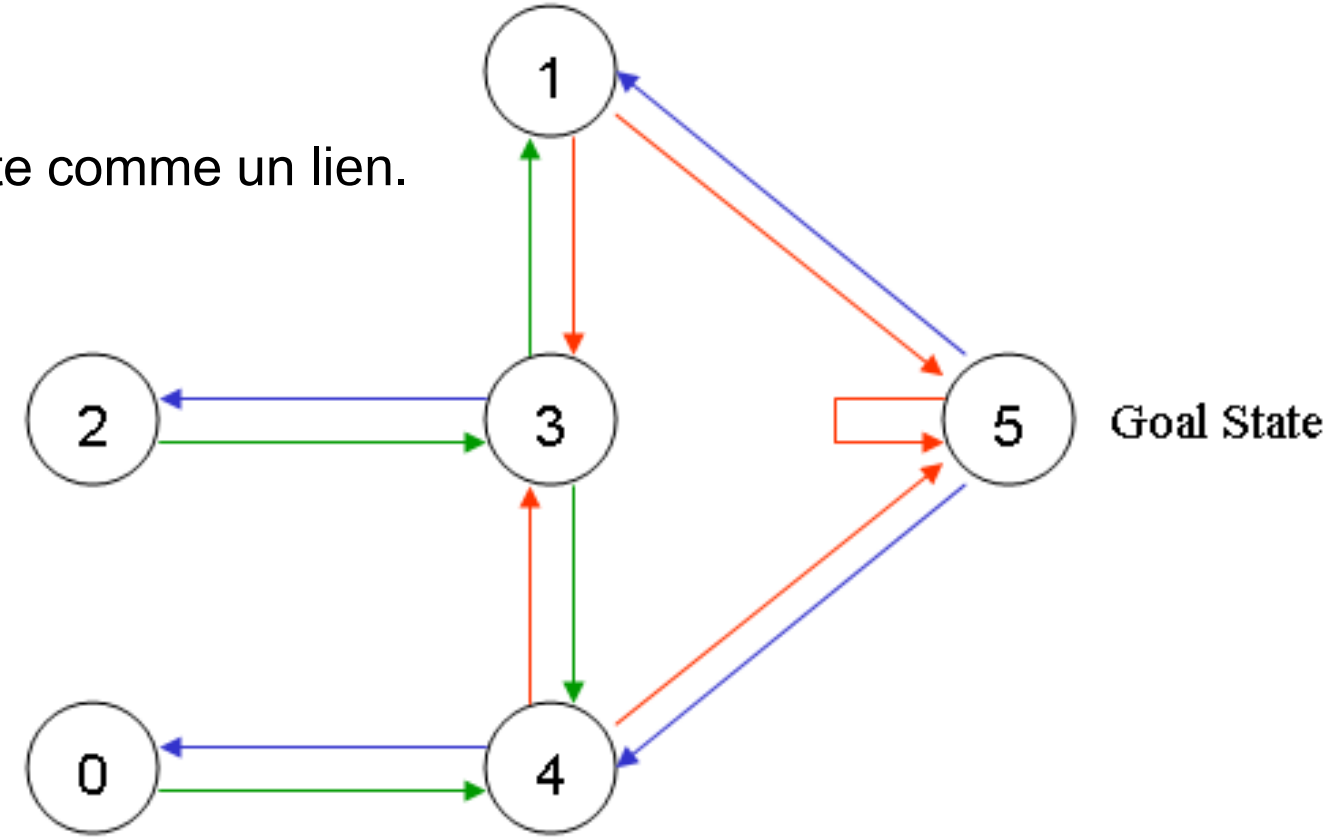
La méthode Qlearning



La méthode Qlearning

La méthode Qlearning

Chaque pièce comme nœud, et chaque porte comme un lien.

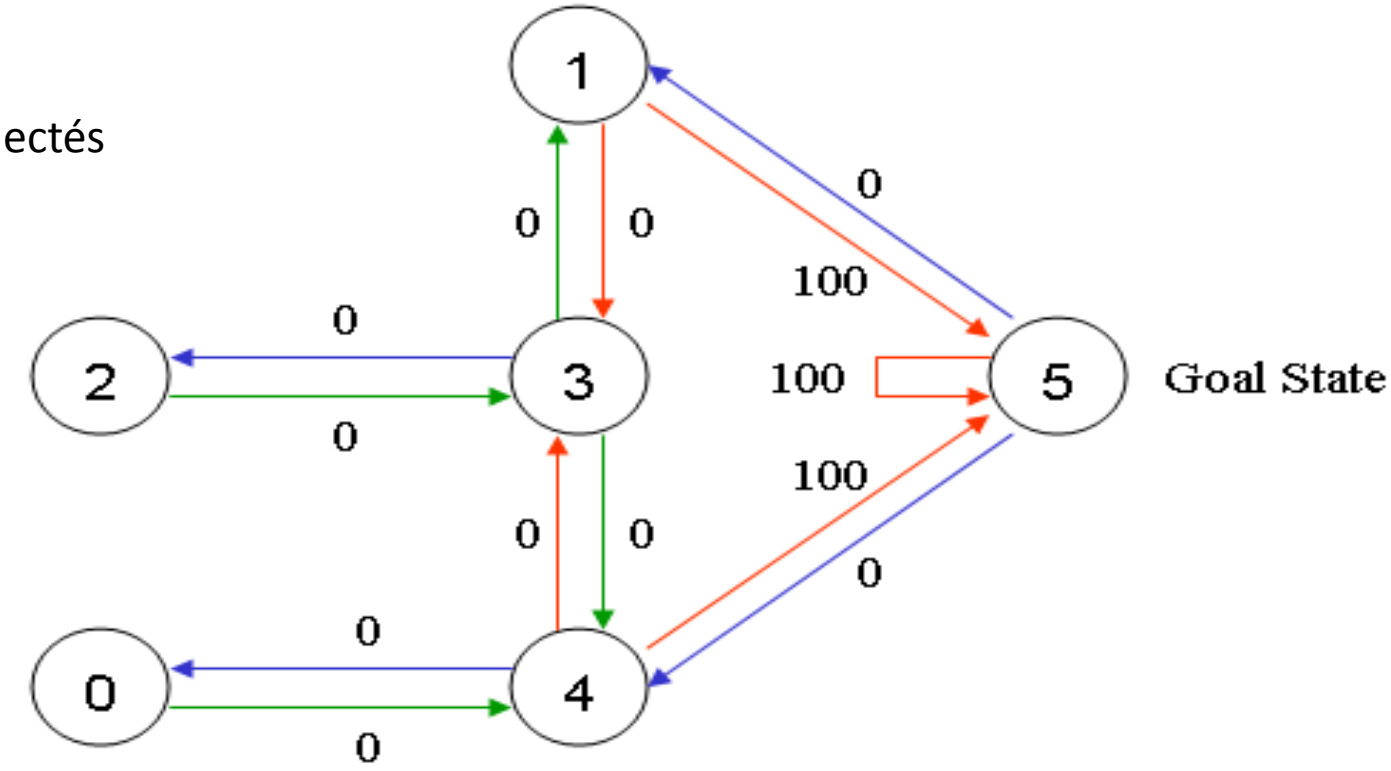


La méthode Qlearning

- Les portes qui mènent immédiatement à l'objectif ont une récompense instantanée de 100

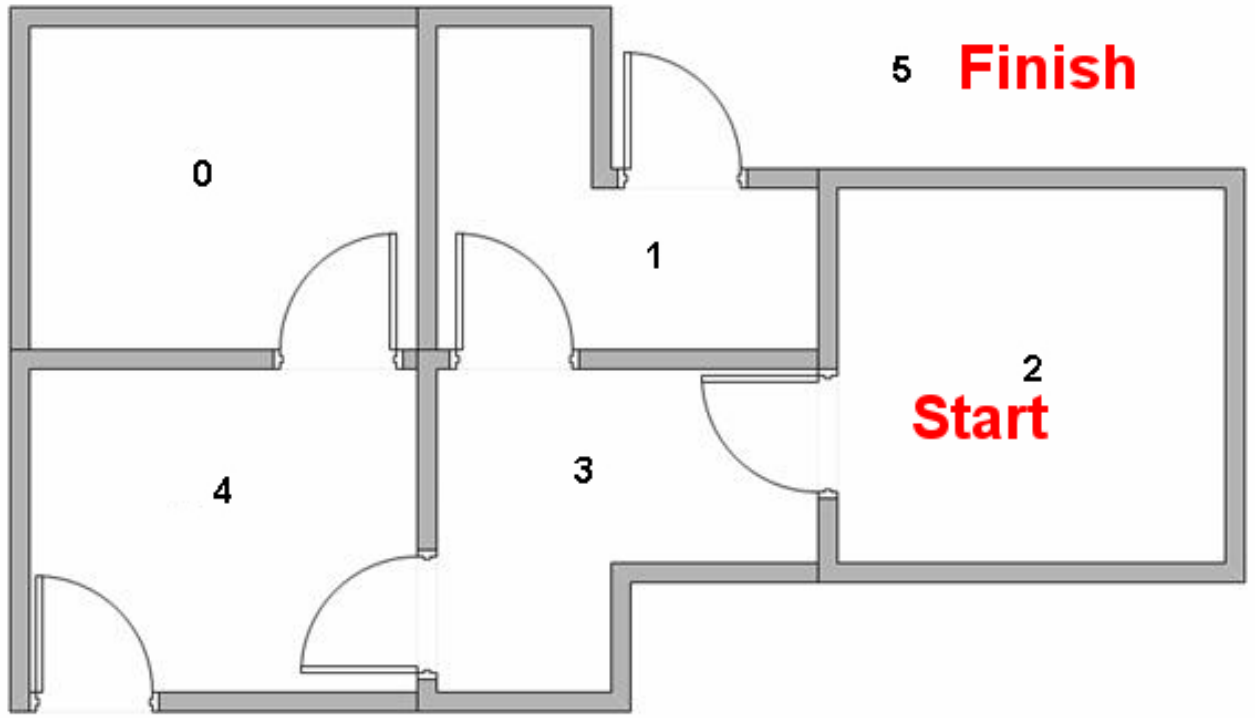
- Les autres portes qui ne sont pas directement connectés à la salle cible ont une récompense nulle

Dans Q-learning, l'objectif est d'atteindre l'état avec la récompense la plus élevée.



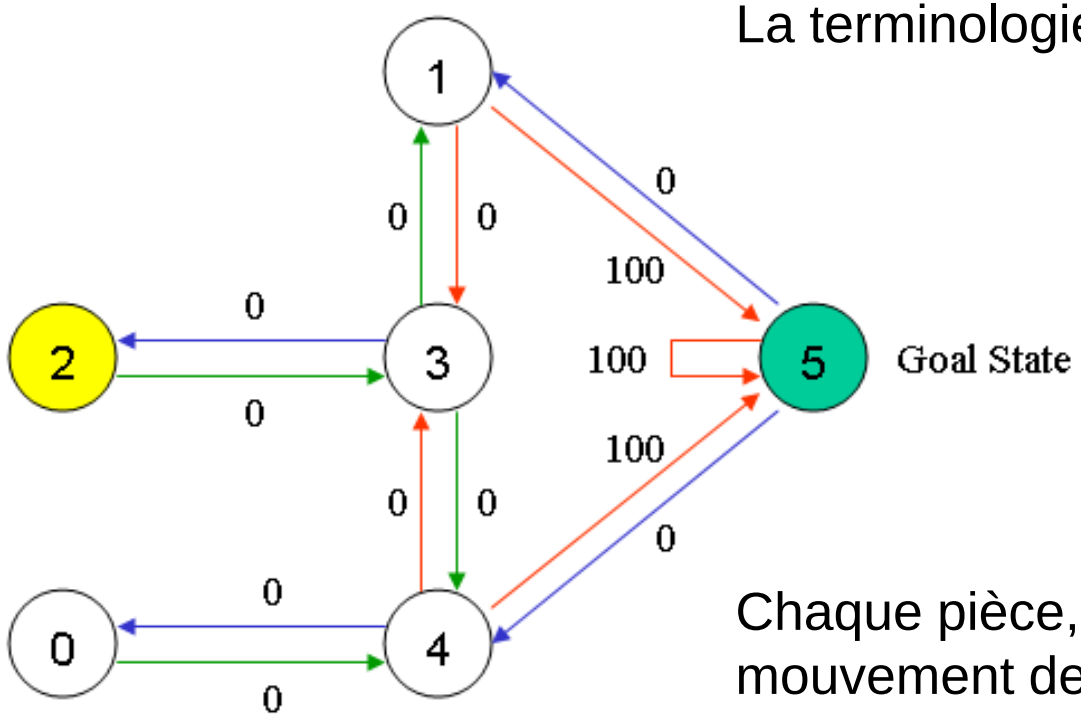
La méthode Qlearning

Imaginons notre agent comme un robot qui veut apprendre par l'expérience.
L'agent peut passer d'une pièce à une autre mais n'a aucune connaissance de l'environnement et ne sait pas quelle séquence de portes mène à l'extérieur



Maintenant, supposons que nous avons un agent dans la salle 2 et nous voulons que l'agent d'apprendre à atteindre la salle 5

La méthode Qlearning

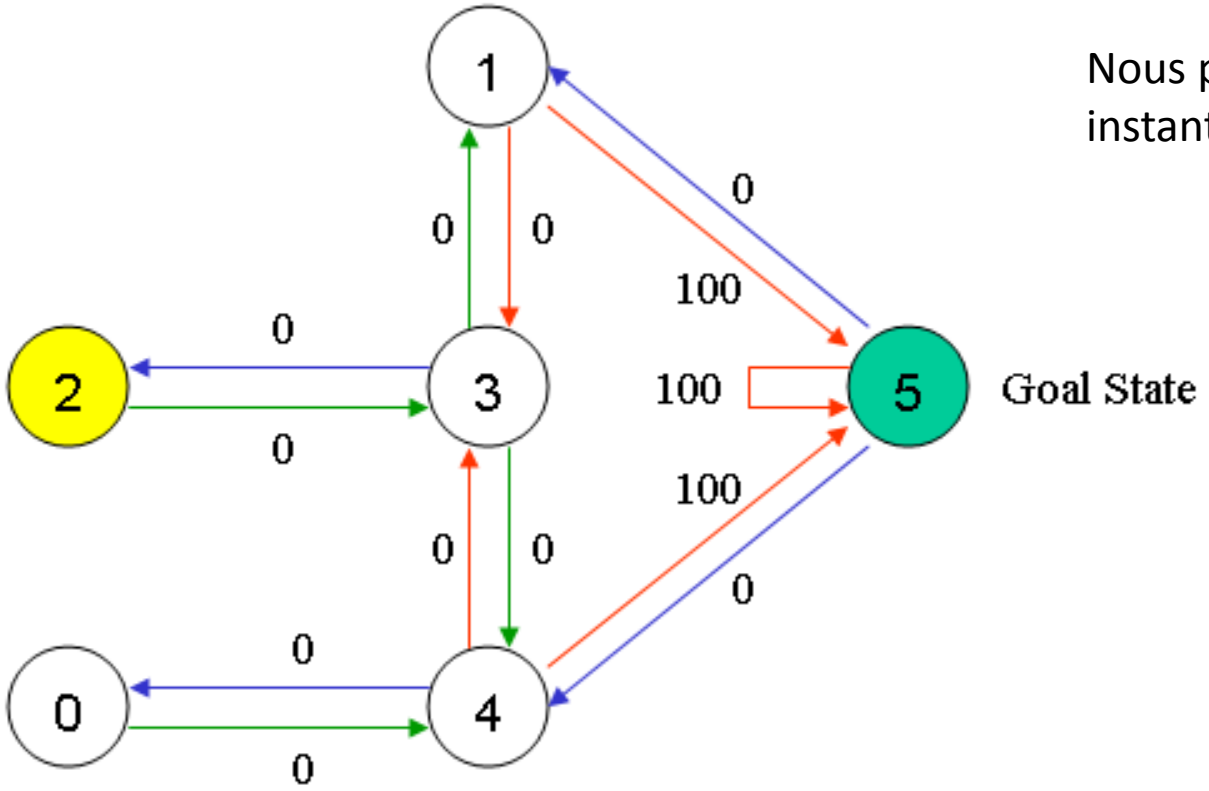


La terminologie de Q-Learning comprend les termes «état» et «action».

un **ensemble d'états** *S* (incluant l'éta initial s_0 et l'état finale Sf)
un **ensemble d'actions** possibles *Actions*(*s*) (ou *A*(*s*)) lorsque je me trouve à l'état *s*

Chaque pièce, y compris l'extérieur est appelée un «état», et le mouvement de l'agent d'une pièce à l'autre sera appelé une «action».

La méthode Qlearning



Nous pouvons mettre le diagramme d'état et les valeurs de récompense instantanée dans la table de récompense suivante :

$$R = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} -1 & -1 & -1 & -1 & 0 & -1 \\ -1 & -1 & -1 & 0 & -1 & 100 \\ -1 & -1 & -1 & 0 & -1 & -1 \\ -1 & 0 & 0 & -1 & 0 & -1 \\ 0 & -1 & -1 & 0 & -1 & 100 \\ -1 & 0 & -1 & -1 & 0 & 100 \end{bmatrix} \end{array}$$

Action

State

Les -1 dans le tableau représentent des valeurs nulles (c'est-à-dire, où il n'y a pas de lien entre les nœuds)

La méthode Qlearning

Maintenant, nous allons ajouter une matrice similaire, «Q», au cerveau de notre agent, représentant la mémoire de ce que l'agent a appris par l'expérience. Les lignes de la matrice Q représentent l'état courant de l'agent et les colonnes représentent les actions possibles menant à l'état suivant

$$Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

$$R = \begin{array}{c} \text{State} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \text{Action} \\ \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} -1 & -1 & -1 & -1 & 0 & -1 \\ -1 & -1 & -1 & 0 & -1 & 100 \\ -1 & -1 & -1 & 0 & -1 & -1 \\ -1 & 0 & 0 & -1 & 0 & -1 \\ 0 & -1 & -1 & 0 & -1 & 100 \\ -1 & 0 & -1 & -1 & 0 & 100 \end{bmatrix} \end{array}$$

L'agent commence à ne rien savoir, la matrice Q est initialisée à zéro

La méthode Qlearning

La règle de transition de Q Learning est une formule très simple:

$$Q(\text{state}, \text{action}) = R(\text{state}, \text{action}) + \text{Gamma} * \text{Max}[Q(\text{next state}, \text{all actions})]$$

Selon cette formule, une valeur affectée à un élément spécifique de la matrice Q est égale à la somme de la valeur correspondante dans la matrice R et du paramètre d'apprentissage Gamma multiplié par la valeur maximale de Q pour toutes les actions possibles dans l'état suivant.

Chaque exploration est un épisode

Dans chaque épisode l'agent se déplace de l'état initial à l'état finale

Chaque fois que l'agent arrive à l'état finale, le programme passe à l'épisode suivant.

La méthode Qlearning

La méthode Qlearning

L'algorithme Q-Learning se déroule comme suit:

1. Set the gamma parameter, and environment rewards in matrix R.
2. Initialize matrix Q to zero.
3. For each episode:

Select a random initial state.

Do While the goal state hasn't been reached.

- Select one among all possible actions for the current state.
- Using this possible action, consider going to the next state.
- Get maximum Q value for this next state based on all possible actions.
- Compute: Q(state, action) = R(state, action) + Gamma * Max[Q(next state, all actions)]
- Set the next state as the current state.

End Do

End For

La méthode Qlearning

La méthode Qlearning

$$R = \begin{array}{c} \\ \text{State} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} & & & \text{Action} & & \\ 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} -1 & -1 & -1 & -1 & 0 & -1 \\ -1 & -1 & -1 & 0 & -1 & 100 \\ -1 & -1 & -1 & 0 & -1 & -1 \\ -1 & 0 & 0 & -1 & 0 & -1 \\ 0 & -1 & -1 & 0 & -1 & 100 \\ -1 & 0 & -1 & -1 & 0 & 100 \end{bmatrix} \end{array}$$

$$Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 80 & 0 \\ 0 & 0 & 0 & 64 & 0 & 100 \\ 0 & 0 & 0 & 64 & 0 & 0 \\ 0 & 80 & 51 & 0 & 80 & 0 \\ 64 & 0 & 0 & 64 & 0 & 100 \\ 0 & 80 & 0 & 0 & 80 & 100 \end{bmatrix} \end{array}$$

1. Set current state = initial state.
2. From current state, find the action with the highest Q value.
3. Set current state = next state.
4. Repeat Steps 2 and 3 until current state = goal state.

La méthode Qlearning

Exemple

- ✓Gamma = 0,8
- ✓L'état initial comme Room 1.
- ✓Initialiser la matrice Q comme matrice nulle.

$$Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

La méthode Qlearning

Exemple

$$R = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} -1 & -1 & -1 & -1 & 0 & -1 \\ -1 & -1 & -1 & 0 & -1 & 100 \\ -1 & -1 & -1 & 0 & -1 & -1 \\ -1 & 0 & 0 & -1 & 0 & -1 \\ 0 & -1 & -1 & 0 & -1 & 100 \\ -1 & 0 & -1 & -1 & 0 & 100 \end{bmatrix} \end{array}$$

Action

State

$$Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

λ = 0,8
Current state : 1
Next State : 5

Q(state, action) = R(state, action) + Gamma * Max[Q(next state, all actions)]

Q(1, 5) = R(1, 5) + 0.8 * Max[Q(5, 1), Q(5, 4), Q(5, 5)] = 100 + 0.8 * 0 = 100

$$Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

La méthode Qlearning

Exemple

$$R = \begin{array}{c} \\ \rightarrow \\ \\ \\ \\ \\ \end{array} \begin{array}{c} \text{Action} \\ \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} \text{State} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{bmatrix} -1 & -1 & -1 & -1 & 0 & -1 \\ -1 & -1 & -1 & 0 & -1 & 100 \\ -1 & -1 & -1 & 0 & -1 & -1 \\ -1 & 0 & 0 & -1 & 0 & -1 \\ 0 & -1 & -1 & 0 & -1 & 100 \\ -1 & 0 & -1 & -1 & 0 & 100 \end{bmatrix} \end{array}$$

$$Q = \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

λ = 0,8
Current state : 3
Next State : 1

Q(state, action) = R(state, action) + Gamma * Max[Q(next state, all actions

Q(3, 1) = R(3, 1) + 0.8 * Max[Q(1, 3), Q(1, 5)] = 0 + 0.8 * Max(0, 100) = 80

$$Q = \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 80 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

La méthode Qlearning

Exemple

$$R = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ \rightarrow\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} \text{Action} & & & & & \\ 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} -1 & -1 & -1 & -1 & 0 & -1 \\ -1 & -1 & -1 & 0 & -1 & 100 \\ -1 & -1 & -1 & 0 & -1 & -1 \\ -1 & 0 & 0 & -1 & 0 & -1 \\ 0 & -1 & -1 & 0 & -1 & 100 \\ -1 & 0 & -1 & -1 & 0 & 100 \end{bmatrix} \end{array} \quad Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 80 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

λ = 0,8
Current state : 1
Next state : 5

$$Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 80 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Q(state, action) = R(state, action) + Gamma * Max[Q(next state, all actions)]

Q(1, 5) = R(1, 5) + 0.8 * Max[Q(5, 1), Q(5, 4), Q(5, 5)] = 100 + 0.8 * Max(0,0,0) = 100

La méthode Qlearning

Exemple

$$R = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} \text{Action} & & & & & \\ 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} -1 & -1 & -1 & -1 & 0 & -1 \\ -1 & -1 & -1 & 0 & -1 & 100 \\ -1 & -1 & -1 & 0 & -1 & -1 \\ -1 & 0 & 0 & -1 & 0 & -1 \\ 0 & -1 & -1 & 0 & -1 & 100 \\ -1 & 0 & -1 & -1 & 0 & 100 \end{bmatrix} \end{array} \quad Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 80 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

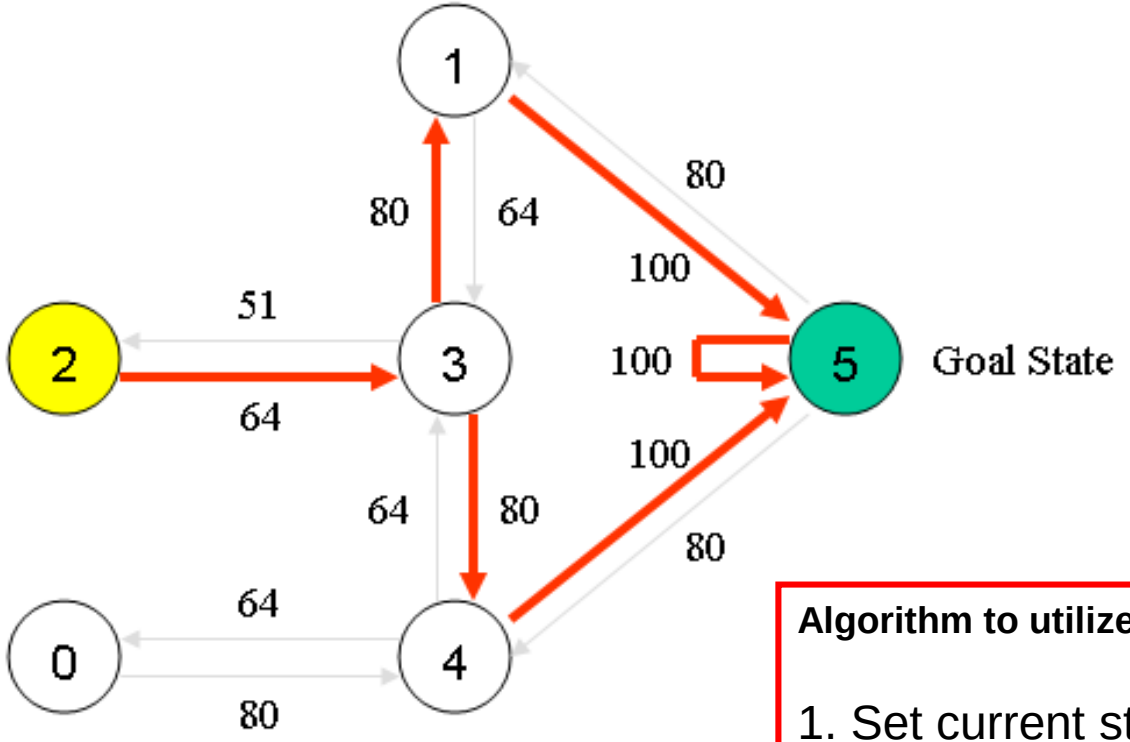
λ = 0,8
Current state : 4
Next state: 3

$$Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 80 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 64 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Q(state, action) = R(state, action) + Gamma * Max[Q(next state, all actions

Q(4, 3) = R(4, 3) + 0.8 * Max[Q(3, 1), Q(3, 2), Q(3, 4)] = 0 + 0.8 * Max(80,0,0) = **64**

$$Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 80 & 0 \\ 0 & 0 & 0 & 64 & 0 & 100 \\ 0 & 0 & 0 & 64 & 0 & 0 \\ 0 & 80 & 51 & 0 & 80 & 0 \\ 64 & 0 & 0 & 64 & 0 & 100 \\ 0 & 80 & 0 & 0 & 80 & 100 \end{bmatrix} \end{array}$$



Algorithm to utilize the Q matrix:

1. Set current state = initial state.
2. From current state, find the action with the highest Q value.
3. Set current state = next state.
4. Repeat Steps 2 and 3 until current state = goal state.

La séquence optimale est :

Mini Projet : Path planning avec la méthode Qlearning