Optimisation des feux rouges
à l'aide de l'intelligence artificielle

Al Hakim TAOUFIK

7 juin 2023

Sommaire

Sommaire

Introduction

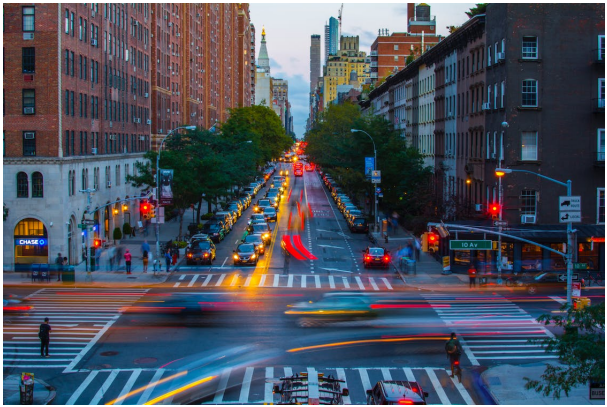


Figure 1 – Congestion du trafic dans une intersection

Problématique

Comment automatiser, optimiser et modéliser les feux tricolores à l'aide de l'intelligence artificielle ?

Sommaire

Procédés de décision Markovien (PDM)

On définit un PDM comme un quadruplet $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$ où :

- $\mathcal{S}$ est l'ensemble d'états fini.
- $\mathcal{A}$ est l'ensemble des actions fini. On note aussi indifféremment la fonction $\mathcal{A} : \mathcal{S} \to \mathscr{P}(\mathcal{A})$.
- $\mathcal{P}$ est la fonction de transition $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$.
- $\mathcal{R}$ est la fonction récompense $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$

Procédés de décision Markovien (PDM)

On considère le temps comme suite discrète $t \in \mathbb{N}$

On considère alors les suites de variables aléatoires discrètes suivantes :

- Suite d'états $(s_t)_{t\in\mathbb{N}}$ à valeurs dans $\mathcal{S}$, suite d'actions $(a_t)_{t\in\mathbb{N}}$ à valeurs dans $\mathcal{A}$
- Suite de récompenses $(r_t)_{t\in\mathbb{N}}$ à valeurs réelles.

On considère aussi la politique stochastique π :

- La fonction $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$

Procédés de décision Markovien (PDM)

Propriété de Markov

Une loi de probabilité $\mathbb{P}$ dans une PDM satisfait la propriété de Markov si :
$\mathbb{P}(s_{t+1}|s_t, a_t, r_t, s_{t-1}, a_{t-1}, ..., r_1, s_0, a_0) = \mathbb{P}(s_{t+1}|s_t, a_t)$

On prend :

$$\forall (s', a, s) \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}, \quad \mathcal{P}(s', a, s) = \mathbb{P}(s_{t+1} = s' | a_t = a, s_t = s)$$

Procédés de décision Markovien (PDM)

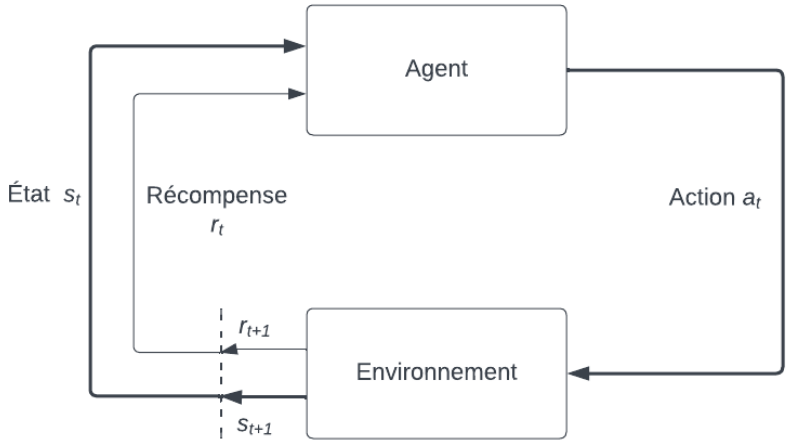


Figure 2 – L'agent en apprentissage interagissant avec son environnement avec une politique π

Fonctions Valeurs

Il faut chercher alors la politique **optimale** π qui **maximise** la somme des récompenses amorties :

$$\mathcal{R}_\infty = r_0 + \gamma r_1 + \gamma^2 r_2 + \gamma^3 r_3 \ldots$$

avec $0 < \gamma < 1$ facteur d'actualisation.

Fonctions Valeurs

On définit la suite de variables aléatoires discrètes réelles $(\mathcal{R}_t)_{t\in\mathbb{N}}$, la récompense cumulée du système :

$$\mathcal{R}_t = \sum_{k=0}^{+\infty} \gamma^k r_{t+k+1}$$

Fonctions Valeurs

Espérance conditionnelle

Soit X une v.a.d. réelle, et Y une v.a.d. Pour tout y appartenant à l'ensemble des issues de Y, si $\mathbb{P}(Y = y) \neq 0$, on peut définir :

$$\mathbb{E}(X|Y = y) = \frac{1}{\mathbb{P}(Y = y)}\mathbb{E}(X.1_{\{Y=y\}})$$

On définit ainsi, presque partout, une variable aléatoire $\varphi(Y)$ définie par :
$\varphi(y) = \mathbb{E}(X|Y = y)$, appelée **espérance de X conditionnée par Y** et notée $\mathbb{E}(X|Y)$, et on a $\mathbb{E}(X) = \mathbb{E}(\varphi(Y)) = \sum_y \varphi(y)\mathbb{P}(Y = y)$.

Fonctions Valeurs

On définit alors la fonction récompense $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ par :

$$\forall (s', a, s) \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}, \quad \mathcal{R}(s', a, s) = \mathbb{E}(r_{t+1} | s_{t+1} = s', a_t = a, s_t = s)$$

Fonctions Valeurs

À chaque instant $t \in \mathbb{N}$, on définit les fonctions valeurs suivantes :

- La fonction valeur, $\mathcal{V}^{\pi} : \mathcal{S} \to \mathbb{R}$ tel que :
$$\mathcal{V}^{\pi}(s) = \mathbb{E}_{\pi}(\mathcal{R}_t | s_t = s)$$

- La fonction action-valeur, $\mathcal{Q}^{\pi} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ tel que :
$$\mathcal{Q}^{\pi}(s, a) = \mathbb{E}_{\pi}(\mathcal{R}_t | s_t = s, a_t = a)$$

Les équations de Bellman

Les équations de Bellman

Les fonctions valeurs peuvent être définies récursivement par :

$$\forall s \in \mathcal{S}, \quad \mathcal{V}^{\pi}(s) = \sum_{a \in \mathcal{A}} \pi(s,a) \sum_{s' \in \mathcal{S}} \mathcal{P}(s',a,s)(\mathcal{R}(s',a,s) + \gamma \mathcal{V}^{\pi}(s))$$

$$\forall s \in \mathcal{S}, \forall a \in \mathcal{A}, \quad \mathcal{Q}^{\pi}(s,a) = \sum_{s' \in \mathcal{S}} \mathcal{P}(s',a,s)(\mathcal{R}(s',a,s) + \gamma \mathcal{V}^{\pi}(s))$$

Les équations de Bellman

Relation d'ordre sur l'ensemble des politiques stochastiques :

$$\pi' \geq \pi \iff \forall s \in \mathcal{S}, \quad \mathcal{V}^{\pi'}(s) \geq \mathcal{V}^{\pi}(s)$$

On définit aussi la fonction valeur optimale $\mathcal{V}^*$:

$$\forall s \in \mathcal{S}, \quad \mathcal{V}^*(s) = \max_{\pi} \mathcal{V}^{\pi}(s)$$

Les équations de Bellman

Le but est de trouver la politique stocastique optimale π^* qui correspond à la politique associée à $\mathcal{V}^*$.

Les équations d'optimalité de Bellman

$$\mathcal{V}^*(s) = \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} \mathcal{P}(s', a, s)(\mathcal{R}(s', a, s) + \gamma \mathcal{V}^*(s'))$$

$$\mathcal{Q}^*(s, a) = \sum_{s' \in \mathcal{S}} \mathcal{P}(s', a, s)(\mathcal{R}(s', a, s) + \gamma \max_{a' \in \mathcal{A}} \mathcal{Q}^*(s', a'))$$

Les équations de Bellman

Algorithme 1 : Itération de la Valeur

Données : $\mathcal{A}, \mathcal{S}, \mathcal{P}, \mathcal{R}, \varepsilon \in \mathbb{R}_+^*$

Initialiser $\mathcal{V}_0(s)$ de manière aléatoire pour tout $s \in \mathcal{S}$, $t = 1$, ;

tant que $\forall s \in \mathcal{S}, \quad |\mathcal{V}_t(s) - \mathcal{V}_{t-1}(s)| > \varepsilon$ **faire**

  pour $s \in \mathcal{S}$ **faire**

    pour $a \in \mathcal{A}$ **faire**

      $\mathcal{Q}_t(s,a) = \sum_{s' \in \mathcal{S}} \mathcal{P}(s',a,s)(\mathcal{R}(s',a,s) + \gamma \max_{a' \in \mathcal{A}} \mathcal{V}_{t-1}(s'))$

    fin

    $\mathcal{V}_t(s) = \max_a \mathcal{Q}_t(s,a)$

  fin

fin

Les équations de Bellman

PROBLÈME : Dans notre modélisation, $\mathcal{R}$ et $\mathcal{P}$ sont inconnues, puisque la loi de probabilité $\mathbb{P}$ est inconnue.

Algorithme d'apprentissage

On utilise l'algorithme du **Q-Learning**, qui consiste à appliquer la formule :

$$\mathcal{Q}(s_t, a_t) \leftarrow \mathcal{Q}(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_{a \in \mathcal{A}} \mathcal{Q}(s_{t+1}, a) - \mathcal{Q}(s_t, a_t)]$$

pour avoir :

$$\forall s \in \mathcal{S}, \forall a \in \mathcal{A}, \quad \lim_{t \to +\infty} \mathcal{Q}_t(s, a) = \mathcal{Q}^*(s, a)$$

Algorithme d'apprentissage

Avantages du Q-Learning :

- Le Q-learning est un algorithme ***sans modèle***.
- Le Q-learning est un algorithme ***off-policy***.

Algorithme d'apprentissage

Algorithme 2 : Algorithme Q-Learning

Données : $\mathcal{A}, \mathcal{S}$, α : taux d'apprentissage, γ : facteur d'actualisation, $\varepsilon \in \mathbb{R}_+^*$
Résultat : Une $\mathcal{Q}$-table contenant les $\mathcal{Q}$-valeurs définissant la politique π^*
Initialiser $\mathcal{Q}(s, a)$ de manière aléatoire pour tout $(s, a) \in \mathcal{S} \times \mathcal{A}$, $i = 1$;
pour *chaque épisode* **faire**
    Initialiser s ;
    pour *chaque étape t de l'épisode* **faire**
        Choisir l'action a en utilisant ***l'algorithme ε-greedy*** ;
        Effectuer l'action a, et recevoir la récompense r, et observer le nouveau état s' ;
        $\mathcal{Q}_{t+1}(s, a) \leftarrow \mathcal{Q}_t(s, a) + \alpha[r + \gamma \max_{a' \in \mathcal{A}} \mathcal{Q}_t(s', a') - \mathcal{Q}_t(s, a)]$, $s \leftarrow s'$
    fin
fin

Algorithme d'apprentissage

Algorithme 3 : Algorithme ε-greedy

Données : $\mathcal{Q}$-table, $s \in \mathcal{S}$, $\varepsilon \in \mathbb{R}_+^*$
Résultat : Action choisie a
$\eta \leftarrow$ nombre aléatoire dans $[0, 1]$;
si $\eta < \varepsilon$ **alors**
  $a \leftarrow$ action aléatoire dans $\mathcal{A}$
sinon
  $a \leftarrow \max_{a' \in \mathcal{A}} \mathcal{Q}(s, a')$
fin
Retourne l'action a sélectionnée

Sommaire

Environnement

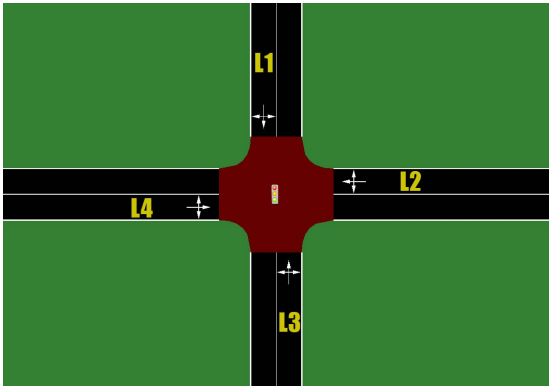
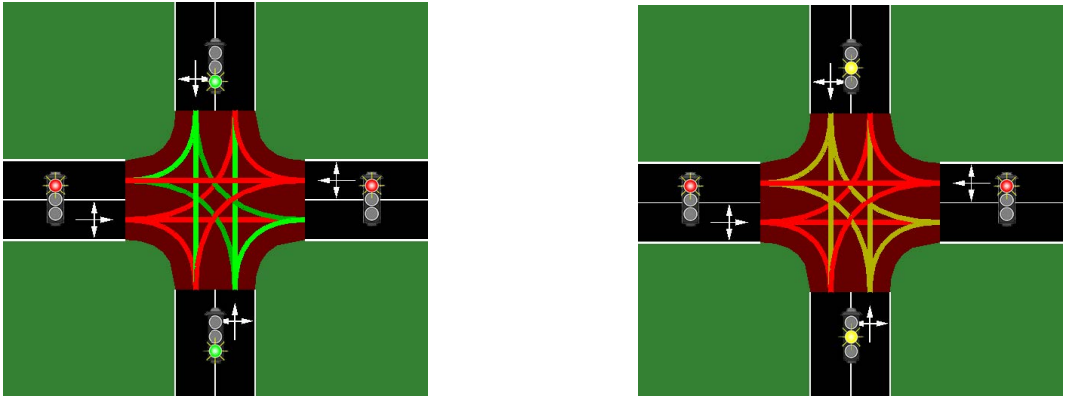


Figure 3 – Carrefour simple

Chaque ligne d'une chaussée est de longueur 100m

Agent

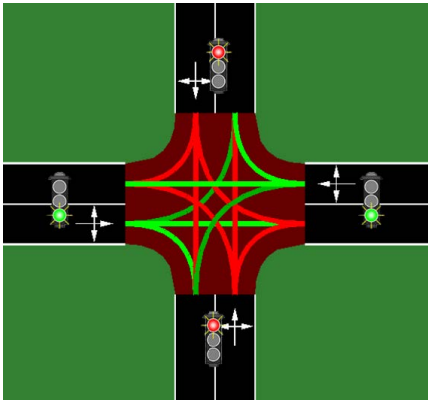


(a) Phase 1

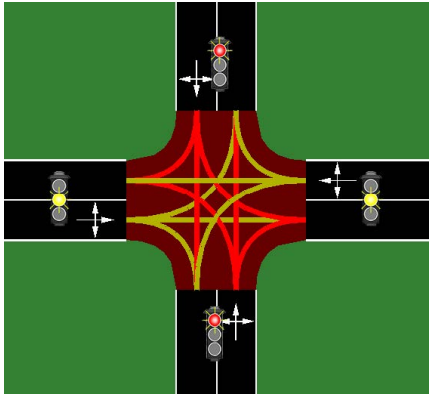
(b) Phase 2

Figure 4 – Les phases de l'agent

Agent



(a) Phase 3



(b) Phase 4

Figure 5 – Les phases de l'agent

Agent

Les actions possibles de notre agent seraient :

- ***Rester*** dans la phase actuelle
- ***Changer*** de phase vers la prochaine

Récompense du système

À chaque étape t, on choisit la récompense du système **à être maximisée** par :

$$r_t = f(t-1) - f(t)$$

avec :

$$f(t) = \sum_{veh=1}^{n} awt(veh, t)$$

et :

$awt(veh, t)$, le temps en secondes écoulé avec $v(veh) < 0.1m/s$ à l'étape t.

Sommaire

Convergence du système

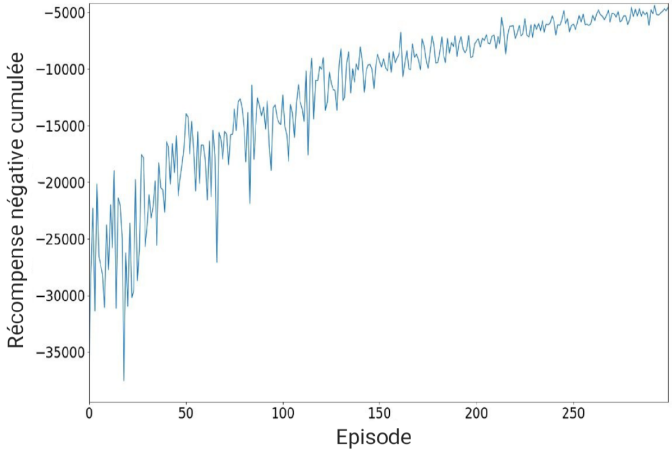


Figure 6 – Récompense du système cumulée par épisode

Performance du système

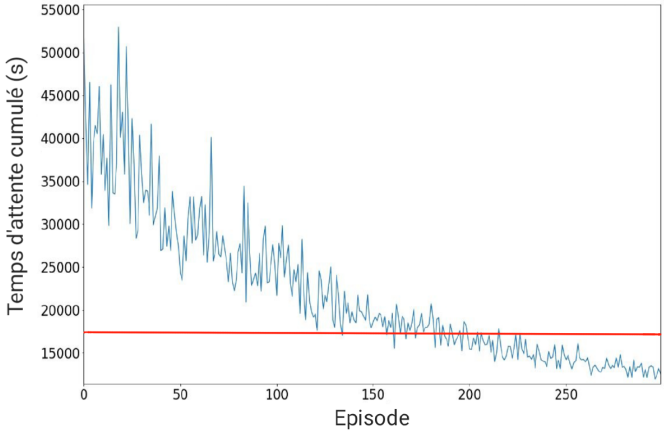


Figure 7 – Temps d'attente cumulé par épisode

Performance du système

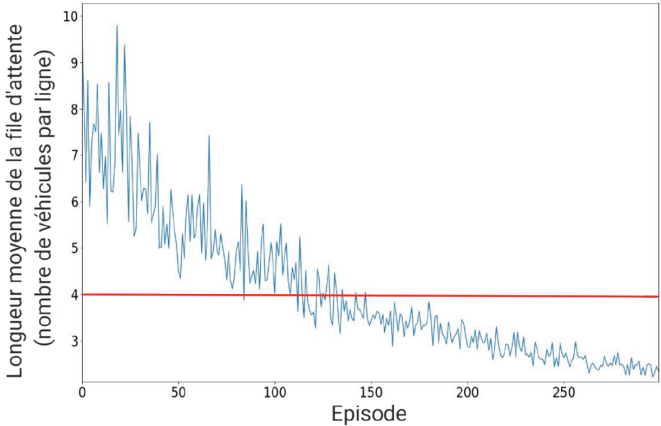


Figure 8 – Longueur moyenne de la file d'attente par épisode

MERCI POUR VOTRE ATTENTION

TAOUFIK AL Hakim