

# Documentation

H4313

3 mai 2016

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Simulateur</b>	<b>2</b>
2.1	Modélisation des tronçons . . . . .	2
2.2	Modèle de gestion du trafic . . . . .	2
2.3	Modèle du changement de voie et file de dépendance . . . . .	3
<b>3</b>	<b>Apprentissage par renforcement</b>	<b>3</b>
3.1	Principes . . . . .	3
3.2	Mise en place dans le projet . . . . .	3
<b>4</b>	<b>Bilan du projet</b>	<b>3</b>
4.1	Tentatives infructueuses . . . . .	3
4.1.1	Deep Learning . . . . .	3
4.2	• . . . .	3
4.3	Tentatives réussies . . . . .	3

# 1 Introduction

Le nombre de véhicules a beaucoup augmenté ces dernières années. Les infrastructures routières se retrouvent alors de plus en plus encombrées. De ce fait, il est nécessaire de réfléchir à une gestion du trafic. De nos jours, les municipalités tentent d'améliorer les transports en commun et les modes de déplacements verts en incitant les automobilistes à ne plus prendre leurs véhicules pour leurs déplacements quotidiens. Cependant, certaines personnes n'ont pas le choix et doivent continuer à prendre leurs véhicules. Nous pensons par exemple aux personnes âgées, aux personnes à mobilité réduite ou bien encore aux livreurs. De plus, le nombre de bus pour les transports en commun participe à l'augmentation du trafic. La nécessité d'une optimisation de la gestion des feux de signalisation nous semblait évidente. C'est pourquoi nous avons décidé de nous pencher sur ce problème très complexe. En effet, plusieurs thèses ont été publiées sur ce sujet<sup>1</sup> et l'optimum du problème n'a toujours pas été trouvé. Nous tenterons, dans le faible temps imparti, de pointer du doigt les points difficiles à l'optimisation des feux de signalisation et de montrer que les méthodes d'apprentissage peuvent être une bonne solution à ce problème.

## 2 Simulateur

Le Grand-Lyon ne possède pas dans son catalogue open-data de données sur les feux de signalisation. En effet, ceux-ci sont gérés par une société privée et la mise à disposition des données de ceux-ci n'est pas une option. Nous avons donc été contraint de développer un simulateur qui reflète du mieux que possible le comportement réel des usagers de la route.

### 2.1 Modélisation des tronçons

Afin de simplifier le problème, nous avons restreints la géométrie des carrefours à des carrefours où les tronçons sont à angles droits et où chaque tronçon dessert au maximum 3 directions (Gauche, Tout Droit, Droite). Chaque voie possède une ou plusieurs directions (Gauche, Tout Droit, Droite) ainsi qu'une vitesse maximale.

Afin de rendre réaliste la trajectoire dans les intersections, nous avons choisi d'interpoler la trajectoire en virage par une courbe d'ordre 3.

### 2.2 Modèle de gestion du trafic

Pour modéliser le comportement des véhicules, deux approches sont possibles. L'approche macroscopique, qui considère un flux de véhicule et effectue des calculs à l'instar de la mécanique des fluides. La deuxième approche, l'approche microscopique, considère les véhicules individuellement et tente de modéliser son comportement, éventuellement en fonction de celui des autres.

Nous avons choisi d'utiliser une approche microscopique et de retenir le modèle du conducteur intelligent (Intelligent Driver Model en anglais<sup>2</sup>). Ce modèle est un des plus récents, un des plus efficaces, et un modèle dans lequel les équations restent compréhensibles et ont un sens physique.

---

1. Régulation du trafic urbain <http://www.biblio.univ-evry.fr/theses/2009/2009EVRY0035.pdf>

2. Page Wikipedia sur IDM [https://en.wikipedia.org/wiki/Intelligent\\_driver\\_model](https://en.wikipedia.org/wiki/Intelligent_driver_model)

Ce modèle donne la vitesse et l'accélération d'un véhicule en fonction de la vitesse maximale de la voie, de la distance avec l'obstacle le précédant et de différents paramètres comme l'accélération maximale ou la décélération maximale (freinage).

### 2.3 Modèle du changement de voie et file de dépendance

## 3 Apprentissage par renforcement

### 3.1 Principes

L'apprentissage par renforcement est une méthode d'apprentissage automatique qui permet d'apprendre à partir d'exemples réalisés dans des situations différentes. Le but est d'optimiser une fonction de récompense. Le

### 3.2 Mise en place dans le projet

Il existe plusieurs bibliothèques d'apprentissage en *Python*. Nous avons choisi la bibliothèque *Pybrain* qui est facile à installer et à utiliser. Il ne propose malheureusement pas de calcul sur GPU contrairement à la bibliothèque *Caffe*, mais celle ci est plus difficile à installer et mettre en œuvre.

La fonction de récompense que nous avons mise en place prend en compte la vitesse moyenne avec une pénalité en fonction du nombre de véhicules arrêtés.

## 4 Bilan du projet

### 4.1 Tentatives infructueuses

#### 4.1.1 Deep Learning

**Principes :** Le Deep Learning est un ensemble de méthode d'apprentissage automatique. Il permet à un ordinateur d'apprendre à partir d'exemples.

**Raison de l'échec :** Le Grand-Lyon n'ayant aucune donnée sur les feux de signalisation, nous n'avons aucun exemple. Nous pouvons construire des exemples à partir de notre simulateur. Cependant, créer suffisamment d'exemples prendra beaucoup trop de temps.

#### 4.1.2 Algorithme génétique

**Principes :** Le principe des algorithmes génétiques est similaire à l'évolution génétique d'un individu. Initialement, plusieurs solutions sont générées et évaluées. A chaque solution est attribué un chance de se reproduire. Avec ses solutions, on crée de nouvelles solutions, composés de gènes provenant de plusieurs solutions mères. On introduit également un phénomène de mutation génétique qui consiste à modifier aléatoire un gène, afin de ne pas tomber dans un optimum local. On répète ce phénomène tant que l'évolution des solutions est visible.

**Raison de l'échec :**