

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université Abderahmane Mira de Béjaia  
Faculté des Sciences Exactes  
Département de Recherche Opérationnelle



## **MÉMOIRE DE FIN DE CYCLE**

EN VUE DE L'OBTENTION DU DIPLÔME DE MASTER EN MATHÉMATIQUES APPLIQUÉES  
Spécialité : Modélisation Optimisation et aide à la décision

Présenté par

**ALLOUACHE HAYAT**

Thème :

---

### **Régulation Adaptative Du Trafic Urbain - Approche par la Théorie des Jeux et L'Apprentissage par Renforcement**

---

Soutenu : le 1/07/2024    Devant le Jury composé de :

Mme	AOUDIA-RAHMOUNE Fazia	Professeur	Présidente
Mr	RADJEF Mohammed Said	Professeur	Rapporteur
Mr	MAHIOUT Hichem	Docteur	Co-Promoteur
Mme	KARA-KENDI Salima	M.C.B	Examinatrice
Mme	DJOUADI-BARACHE Fatiha	M.A.A	Examinatrice

**Année Universitaire : 2023/2024**

# Remerciements

*Je remercie d'abord Dieu le Tout Puissant pour m'avoir donné la force et le courage d'accomplir ce travail.*

*Je tiens à exprimer ma profonde gratitude à Monsieur Mohammed Said RADJEF , Professeur à l'Université de Béjaïa, et à Monsieur Hicham MAHIOUT , Docteur à l'Université de Béjaïa, pour l'honneur qu'ils m'ont fait en étant les rapporteurs de mon mémoire de Master. Je les remercie pour leurs grandes qualités humaines, leur précieuse attention, et particulièrement pour leurs conseils et orientations qui ont grandement contribué à l'aboutissement de ce travail.*

*Mes sincères remerciements et ma gratitude vont également à Madame AOUDIA-RAHMOUNE Fazia, Professeure à l'Université de Béjaïa, pour avoir accepté de juger ce travail et de présider le jury de soutenance. Soyez assurée de mon entière reconnaissance.*

*Je remercie également les autres membres du jury , Madame KARA-KENDI salimaet Madame DJOUADI-BARACHE Fatiha d'avoir bien voulu donner de leurs temps pour lire ce mémoire et d'avoir juger mon travail*

*Un grand merci à tous les enseignants du Département de Recherche Opérationnelle de l'Université de Béjaïa pour leur dévouement et leur soutien tout au long de ma formation universitaire.*

*J'ai une pensée particulière pour ma chère famille et ma chère petite amie. Leur soutien indéfectible et leurs encouragements constants ont été pour moi une source inestimable de force et de motivation tout au long de mes études. Leur présence et leur soutien inconditionnel ont été essentiels pour surmonter les défis et atteindre mes objectifs académiques. Je leur suis profondément reconnaissante pour tout ce qu'ils ont fait pour moi.*

*Merci aussi à tous mes amis, mes camarades et toutes les personnes que j'ai pu côtoyer pendant ces cinq ans à l'université. Leurs encouragements, leur camaraderie et les moments partagés ont enrichi mon expérience académique et personnelle. Leur soutien et leur amitié ont été une bouée précieuse dans les périodes de doute et de stress. Je leur exprime ma profonde sympathie et je leur souhaite sincèrement le meilleur pour leurs propres projets et aspirations.*

*Enfin, merci à toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce mémoire. Leur expertise et leur soutien moral ont été essentiels à l'aboutissement de ce travail.*

# Dédicaces

*“À toutes les personnes qui m’aiment, Merci pour votre soutien indéfectible et votre amour constant, qui ont été ma source de force et d’inspiration.”*

*Allouache. H* 

# Table des Matières

<b>TABLE DES MATIÈRES</b>	<b>II</b>
<b>LISTE DES TABLEAUX</b>	<b>V</b>
<b>LISTE DES FIGURES</b>	<b>VI</b>
<b>LISTE DES ABRÉVIATIONS</b>	<b>VIII</b>
<b>INTRODUCTION GÉNÉRALE</b>	<b>1</b>
<b>1 SYSTEMES DE TRANSPORT INTELLIGENTS</b>	<b>3</b>
1.1 Le trafic routier . . . . .	3
1.2 Le réseau routier . . . . .	4
1.2.1 Routes locales . . . . .	4
1.2.2 Routes nationales et régionales . . . . .	4
1.2.3 Autoroutes et voies rapides . . . . .	5
1.3 Intersections . . . . .	5
1.4 Congestion routière . . . . .	6
1.5 Feu de circulation tricolore . . . . .	7
1.6 Système de gestion des feux tricolores . . . . .	8
1.6.1 Fonctionnement cyclique des feux de signalisation . . . . .	8
1.6.2 L'évolution des systèmes de gestion des feux tricolores . . . . .	9
1.6.3 La signalisation intelligente . . . . .	10
1.7 Architecture des feux de signalisation intelligentes . . . . .	10
1.7.1 Module de collecte des statistiques de trafic . . . . .	10
1.7.2 Module de contrôle du trafic . . . . .	11
1.8 Logiciels de gestion et de commande des feux de signalisation . . . . .	12
<b>2 OPTIMISATION DU CONTRÔLE DES FEUX DE SIGNALISATION : APPROCHES MATHÉ-</b>	
<b>MATIQUES</b>	<b>14</b>
2.1 Exploration des Recherches Précédentes . . . . .	14
2.1.1 La Programmation Dynamique (PD) . . . . .	15

---

2.1.2	Les heuristiques et les métaheuristiques . . . . .	15
2.1.3	La théorie des jeux . . . . .	15
2.2	La théorie des jeux et l'optimisation linéaire . . . . .	16
2.2.1	La programmation linéaire (PL) . . . . .	16
2.2.2	La théorie des jeux . . . . .	17
2.2.3	Les Concepts Clés de la TJ . . . . .	17
2.2.4	Présentation d'un jeu sous forme normale . . . . .	18
2.2.5	Méthodes de Résolution en Théorie des Jeux . . . . .	18
2.3	Problématique . . . . .	20
2.4	Présentation d'une intersection . . . . .	20
2.5	Modélisation du Trafic . . . . .	22
2.6	Résolution . . . . .	25
<b>3</b>	<b>OPTIMISATION DU CONTRÔLE DES FEUX DE SIGNALISATION : APPROCHES DE</b>	
	<b>L'INTELLIGENCE ARTIFICIELLE</b>	<b>28</b>
3.1	Les techniques découlant de l'intelligence artificielle . . . . .	28
3.1.1	Les algorithmes génétiques . . . . .	28
3.1.2	Les réseaux de neurones artificiels (ANN) . . . . .	29
3.1.3	Systèmes Experts (SE) . . . . .	29
3.1.4	Reinforcement Learning (RL) . . . . .	29
3.1.5	Algorithmes d'Apprentissage Automatique (ML) . . . . .	30
3.2	Exploration des Paradigmes de l'Intelligence Artificielle : Machine Learning, Deep Learning et Apprentissage par Renforcement . . . . .	31
3.3	Deep Learning (Apprentissage profond) : . . . . .	32
3.3.1	Introduction aux réseaux de neurones artificiels (ANN) . . . . .	32
3.3.2	Architecture des réseaux de neurones . . . . .	32
3.3.3	Les différents types de réseaux neuronaux . . . . .	33
3.3.4	Les méthodes d'apprentissage du réseau neuronal artificiel . . . . .	34
3.3.5	Les composants fondamentaux d'un neurone artificiel . . . . .	34
3.3.6	L'entraînement des réseaux de neurones artificiels . . . . .	35
3.3.7	Introduction aux Fonctions d'Activation . . . . .	36
3.3.8	Les algorithmes dans les réseaux de neurones artificiels . . . . .	37
3.3.9	Intégration du Biais . . . . .	38
3.3.10	Gestion de l'Erreur . . . . .	38
3.4	Reinforcement Learning (Apprentissage par renforcement) . . . . .	39
3.4.1	Composants Principaux . . . . .	39
3.4.2	L'Algorithmes d'apprentissage par renforcement . . . . .	40
3.5	Problématique . . . . .	42
3.6	L'apprentissage par renforcement(RL) et la théorie des jeux (TJ) . . . . .	43

---

---

3.7	Principe général du jeu avec RL proposé . . . . .	43
3.8	Le Processus d'Apprentissage . . . . .	47
3.9	Séparation des Q-Tables par Agent . . . . .	48
<b>4</b>	<b>Simulation et Implémentation</b>	<b>50</b>
4.1	Présentation du logiciel SUMO . . . . .	50
4.2	Composants des Scénarios dans SUMO . . . . .	51
4.3	Construction d'un Réseau Routier sur SUMO avec un Fichier XML . . . . .	52
4.4	Présentation de TraCI . . . . .	54
4.5	Implémentation du contrôle MetMod . . . . .	54
4.5.1	Interaction entre les Véhicules et les Contrôleurs d'Intersection . . . . .	56
4.5.2	Comparaison des performances globales du réseau des stratégies MetMod 2-phase, MetMod 4-phase et Cournot. . . . .	56
4.6	Implémentation du contrôle MetMod-RL . . . . .	58
4.6.1	Paramètres du MetMod-RL . . . . .	58
4.6.2	Comparaison des performances globales du réseau des stratégies MetMod-RL 4-phase et MetMod 4-phase . . . . .	59
4.6.3	Interprétation des résultats . . . . .	59
4.7	Comparaison des performances de stratégie MetMod-RL . . . . .	60
4.7.1	Paramètres en cas de la présence des collecteurs de données . . . . .	60
4.7.2	Paramètres en cas de l'absence des collecteurs de données . . . . .	61
4.7.3	Interprétation des résultats . . . . .	61
	<b>Bibliographie</b>	<b>64</b>

# Liste des Tableaux

4.1 Flux de véhicules à l'intersection par période . . . . .	55
4.2 Paramètres de simulation de la stratégie MetMod à 2-phases . . . . .	55
4.3 Paramètres de simulation de la stratégie MetMod à 4-phases . . . . .	56
4.4 Paramètres de simulation de la stratégie MetMod-RL à 4-phases . . . . .	58
4.5 Paramètres de simulation de la stratégie MetMod-RL à 4-phases . . . . .	58
4.6 Résultats de la Figure 4.12 . . . . .	59
4.7 Paramètres de simulation de la stratégie MetMod-RL à 4-phases . . . . .	61

# Liste des Figures

1.1	<i>Image du trafic routier</i>	4
1.2	<i>Image d'une intersection</i>	5
1.3	<i>Les formes d'une intersection</i>	6
1.4	<i>Intersection contrôlées</i>	6
1.5	<i>Intersection non contrôlées</i>	6
1.6	<i>Les feu tricolore</i>	7
1.7	<i>Plan d'un cycle composé de quatre phases [44]</i>	9
1.8	<i>Les différents capteurs [45]</i>	11
1.9	<i>Structure de la signalisation intelligente.</i>	11
1.10	<i>L'architecture de TRANSYT</i>	12
1.11	<i>SCOOT</i>	12
2.1	<i>Exemple d'une intersection T.</i>	21
2.2	<i>Phases typiques d'une intersection [3]</i>	21
3.1	<i>Le monde de l'intelligence artificielle</i>	31
3.2	<i>Un neurone dans le cerveau humain</i>	32
3.3	<i>Un neurone dans le cerveau humain</i>	33
3.4	<i>Les composant fondamental d'un neurone artificielle</i>	35
3.5	<i>Composants Q-learning</i>	39
4.1	<i>Logo logiciel SUMO</i>	50
4.2	<i>Configuration des nœuds</i>	52
4.3	<i>Configuration des arêtes</i>	53
4.4	<i>Configuration des types</i>	53
4.5	<i>Combinaison des fichiers XML avec netconvert</i>	53
4.6	<i>Génération du Trafic Routier</i>	53
4.7	<i>Génération du Trafic Routier</i>	54
4.8	<i>Réseau de quatre intersections contrôlées par des feux de signalisation</i>	55
4.9	<i>Nombre de véhicules insérés dans le réseau</i>	56
4.10	<i>Temps moyen de trajet dans le réseau</i>	57

4.11 <i>Nombre de véhicules en circulation</i> . . . . .	58
4.12 Comparaison des Performances des Simulations de Contrôle MetMod-RL et MetMod-4phase. . . . .	59
4.13 Comparaison des Performances des Simulations de Contrôle MetMod-RL-ANN et MetMod-RL. . . . .	61

# Liste des Abréviations

- **STI** : Système de Transport Intelligent (Intelligent Transportation System).
- **TRANSYT** : Traffic Network Study Tool.
- **SCOOT** : Split Cycle and Offset Optimization Technique.
- **PRODYN** : PROgrammation DYNamique.
- **CERT** : Centre d'Études et de Recherche de Toulouse.
- **GTMH** : Groupement Technique des Moyens d'Habitat.
- **PD** : Programmation Dynamique .
- **TJ** : Théorie des Jeux .
- **GNE** : Generalized Nash Equilibrium .
- **LGNEPSC** : Logarithmic Generalized Nash Equilibrium Problem with Shared Constraints .
- **KKT** : Conditions de Karush-Kuhn-Tucker.
- **PL** : Programmation Linéaire.
- **IA** : Intelligence Artificielle .
- **ANN** : Artificial Neural Network.
- **SE** : Systèmes Experts .
- **RL** : Reinforcement Learning .
- **ML** : Machine Learning.

# Introduction Générale

Depuis plusieurs décennies, l'augmentation du niveau de vie a entraîné une croissance significative des déplacements urbains, provoquant une hausse du nombre de véhicules en circulation et une dégradation de la qualité de vie due à la congestion. Pour y remédier, diverses solutions ont été proposées, telles que les sens giratoires, les barrages, les sens uniques et les autoroutes. Cependant, la régulation du trafic par les feux de signalisation reste la solution la plus adoptée. Les feux de signalisation urbains se divisent en deux catégories : fixes et adaptatifs. Les feux fixes, basés sur des cycles prédéterminés, sont simples à mettre en œuvre mais inefficaces face aux variations du trafic en temps réel. À l'inverse, les feux adaptatifs ajustent leurs cycles grâce à des capteurs et des technologies de surveillance, offrant une gestion proactive et flexible du trafic. Bien qu'ils nécessitent des investissements plus importants, les feux adaptatifs améliorent considérablement la fluidité du trafic, réduisent les temps d'attente, réduisent la congestion et améliorent la sécurité routière.

Ce projet explore le développement d'un système de contrôle intelligent des feux de circulation en intégrant la théorie des jeux, l'apprentissage automatique et l'apprentissage par renforcement, dans le but d'atténuer la congestion, d'optimiser le flux de trafic et de réduire les impacts négatifs associés.

Après une introduction générale, le reste de ce mémoire est structuré autour de quatre chapitres, une conclusion générale et s'achève par une bibliographie.

**Chapitre 1 :** Ce chapitre est consacré aux Systèmes de Transport Intelligents (STI), avec un accent particulier sur la gestion et le contrôle des feux de circulation dans un contexte urbain. Il examine divers aspects des infrastructures routières et des technologies de gestion du trafic, en s'intéressant particulièrement aux feux de signalisation et aux stratégies de régulation du trafic.

**Chapitre 2 :** Ce chapitre explore deux parties cruciales pour la gestion optimale des feux de signalisation :

Dans la première partie, l'accent est mis sur la structuration des mouvements de trafic à travers des phases et cycles définis. Chaque intersection est modélisée comme un graphe orienté, où les sommets représentent les intersections et les arcs représentent les mouvements de trafic. La gestion efficace du trafic nécessite l'évaluation précise de la demande de trafic, incluant le

nombre de véhicules, les débits de sortie et d'arrivée.

La seconde partie reprend une approche mathématique basée sur la théorie des jeux, développée dans la thèse de doctorat H.Mahiout [30]. Les résultats de cette approche sont synthétisés sous forme d'un d'un algorithme, applé MetMod, qui permet de contrôler des feux de signalisation dans un réseau urbain .

**Chapitre 3 :** Ce chapitre se concentre sur l'optimisation du contrôle des feux de signalisation en utilisant des approches basées sur l'intelligence artificielle, en particulier l'apprentissage par renforcement. Il introduit des concepts tels que le Q-learning et les méthodes de Policy Gradient pour permettre à un agent (décideur) d'apprendre à prendre des décisions séquentielles dans un environnement dynamique, afin de maximiser les récompenses. Ces approches visent à améliorer la gestion du trafic urbain en adaptant les durées de feux verts en temps réel, réduisant ainsi les congestions et améliorant l'efficacité du système de signalisation urbaine.

**Chapitre 4 :** Ce chapitre est dédié à l'implémentation et à la simulation des stratégies de contrôle de trafic MetMod. Nous présentons les résultats obtenus en simulant une stratégie de contrôle de feux de signalisation dans un réseau routier urbain avec des intersections à 12 mouvements, dans le cas où ils sont répartis en deux phases, puis dans le cas où ils sont répartis en quatre phases. Par la suite, nous avons simulé le modèle MetMod à quatre phases après avoir développé dans ce mémoire une stratégie MetMod-RL et des réseaux de neurones (MetMod-RL-ANN). Nous terminons par une analyse comparative entre les différentes variantes et avec un modèle de cournot [24].

# Chapitre 1

# SYSTEMES DE TRANSPORT INTELLIGENTS

## Introduction

L'infrastructure routière constitue la toile fondamentale sur laquelle repose le mouvement des personnes, des marchandises et des services dans une région donnée. De la simple route locale aux autoroutes interurbaines, chaque élément contribue à façonner la manière dont nous naviguons dans notre environnement quotidien.

Dans ce chapitre, nous explorerons les différents aspects de l'infrastructure routière, des routes elles-mêmes aux intersections complexes, en examinant leur rôle crucial dans la gestion du trafic et les défis auxquels elles sont confrontées.

## 1.1 Le trafic routier

Le trafic routier est l'ensemble des mouvements de véhicules sur les voies de circulation, qu'elles soient urbaines, périurbaines ou rurales. Il englobe non seulement les véhicules motorisés (*voitures particulières, camions, autobus, motos*), mais aussi les modes de transport non motorisés (*vélos, piétons*). Le trafic routier peut être influencé par de nombreux facteurs, tels que l'heure de la journée, les conditions météorologiques, les événements spéciaux, les accidents et les travaux routiers. Il est souvent mesuré en termes de densité de circulation, de vitesse moyenne, de temps de trajet et de nombre de véhicules par heure. Il est crucial de gérer efficacement le trafic routier afin de garantir la sécurité des utilisateurs de la route, diminuer la congestion, réduire les temps de trajet et préserver l'environnement. Cela nécessite fréquemment l'emploi de méthodes comme la signalisation routière, les feux de circulation, les créneaux

urbains, la planification des transports et les technologies de gestion du trafic.



FIGURE 1.1 – *Image du trafic routier*

## **1.2 Le réseau routier**

Un réseau routier est un système complexe de routes, de voies de circulation et d'infrastructures connexes qui permettent le déplacement des personnes, des marchandises et des services sur un territoire donné. Le réseau routier comprend différents types de routes, allant des petites routes locales aux grandes autoroutes interurbaines. Il peut être subdivisé en plusieurs catégories, notamment les routes locales , nationales ou régionales , les autoroutes et les voies rapides.

### **1.2.1 Routes locales**

Les routes locales permettent de relier les quartiers résidentiels, les entreprises locales et les zones commerciales. En général, elles ont une capacité limitée et sont fréquemment employées pour des déplacements ciblés à l'intérieur des villes et des villages.

### **1.2.2 Routes nationales et régionales**

Ces voies permettent la connexion entre les villes, les régions et les États. Elles sont destinées à des trajets plus longs et ont une capacité supérieure à celle des routes locales.

### 1.2.3 Autoroutes et voies rapides

Ce sont des routes à accès limité conçues pour permettre une circulation rapide et fluide des véhicules sur de longues distances. Elles sont souvent équipées de chaussées séparées, de bretelles d'accès et de sortie, échangeurs et d'autres aménagements spécifiques pour assurer la sécurité et la fluidité de la circulation.

## 1.3 Intersections

Une intersection est un point où deux ou plusieurs routes se croisent ou se rejoignent. C'est un lieu où les véhicules peuvent changer de direction ou traverser d'autres voies de circulation.



FIGURE 1.2 – Image d'une intersection

Les intersections peuvent prendre différentes formes, telles que :

1. **Intersection à quatre voies (ou croisement en T) :** Il s'agit d'une intersection où deux routes principales se croisent à angle droit, formant une configuration en forme de T. Cette intersection peut être contrôlée par des feux de signalisation ou des panneaux de priorité.
2. **Intersection en croix (ou intersection en X) :** C'est une intersection où deux routes se croisent formant une croix. Les intersections en croix peuvent être contrôlées par des feux de signalisation ou non, selon le volume de trafic et la sécurité requise.

3. **Intersection en Y** : C'est une intersection où une route se divise en deux branches, formant une configuration en forme de Y. Les intersections en Y peuvent être contrôlées par des feux de signalisation ou non, en fonction des besoins de circulation.
4. **Intersection en \*** : Une intersection en étoile est une intersection complexe avec de multiples directions possibles pour les conducteurs, rendant son franchissement plus difficile. Il est crucial d'être très attentif à la signalisation routière, qu'elle soit horizontale ou verticale.



FIGURE 1.3 – Les formes d'une intersection

Les intersections peuvent être contrôlées ou non contrôlées. Les intersections contrôlées sont équipées de feux de circulation, de panneaux de signalisation, ou de marquages au sol pour réguler le flux de trafic. Les intersections non contrôlées n'ont pas de dispositifs de signalisation et la priorité de passage est déterminée par les règles de la route.



FIGURE 1.4 – Intersection contrôlées

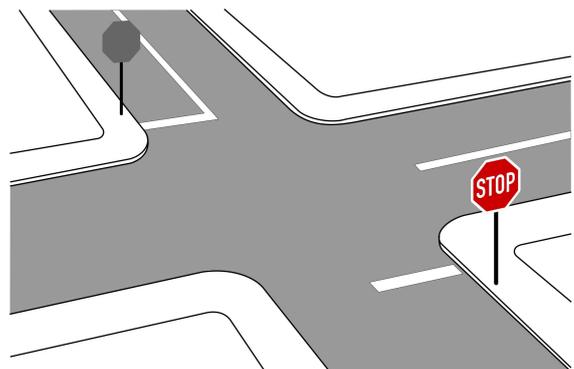


FIGURE 1.5 – Intersection non contrôlées

## 1.4 Congestion routière

La congestion routière se produit lorsque la demande de déplacements dépasse la capacité des infrastructures routières d'une zone urbaine, entraînant des embouteillages, des retards importants et une circulation lente. Ce phénomène est particulièrement ressenti durant les heures

de pointe, lorsque de nombreux conducteurs se rendent au travail ou rentrent chez eux. Bien que diverses mesures soient mises en œuvre par les autorités pour atténuer ce problème, la congestion reste un défi majeur dans de nombreuses villes à travers le monde [32].

On distingue deux types de congestions [43] :

1. **Congestion récurrente** : La congestion récurrente se produit de manière régulière et prévisible aux heures de pointe en raison d'une capacité d'infrastructure insuffisante pour répondre à la demande de trafic. Elle est associée à des problèmes structurels et peut être attribuée à un sous-dimensionnement de l'infrastructure par rapport à la demande réelle, même si cette capacité satisfait un certain niveau de demande à d'autres moments.
2. **Congestion non récurrente** : La congestion non récurrente est causée par des perturbations temporaires sur le réseau routier, telles que des accidents, des travaux routiers ou des conditions météorologiques défavorables. Contrairement à la congestion récurrente, elle est imprévisible et peut varier en intensité et en durée. Ces perturbations peuvent entraîner des ralentissements soudains et importants sur les routes.

## 1.5 Feu de circulation tricolore

Un feu de circulation tricolore est un dispositif de signalisation routière automatisé utilisé aux intersections pour contrôler et réguler le flux de circulation des véhicules et des piétons. Il se compose de trois couleurs principales [44] : le rouge, le jaune et le vert. Chaque couleur a une signification spécifique pour les conducteurs et les piétons.

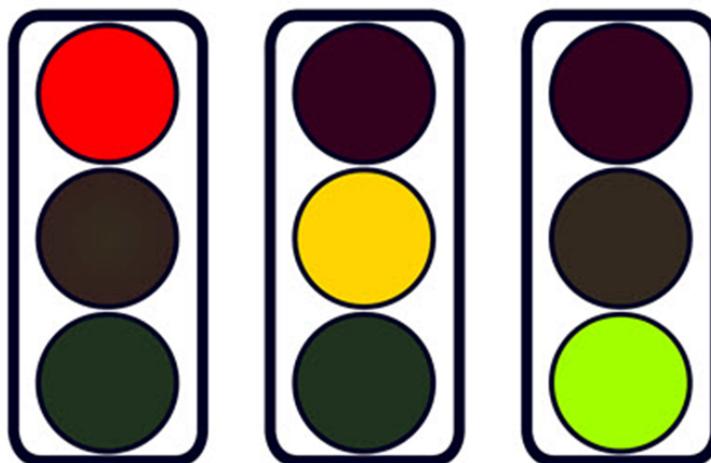


FIGURE 1.6 – Les feux tricolores

1. **Le rouge** : indique l'arrêt complet des véhicules. C'est le signal pour les conducteurs de s'arrêter et d'attendre.

2. **Le jaune** : est généralement utilisé comme avertissement, indiquant que le feu va passer du rouge au vert. C'est un signal pour les conducteurs de se préparer à démarrer.
3. **Le vert** : signifie que les véhicules ont la permission de traverser l'intersection en toute sécurité. C'est le signal pour les conducteurs de reprendre leur route.

Les feux tricolores sont souvent équipés de flèches directionnelles pour indiquer des mouvements spécifiques autorisés, tels que tourner à gauche ou à droite. Les feux destinés aux piétons peuvent être bicolores et affichent généralement une silhouette de piéton pour indiquer quand traverser en toute sécurité. Ces feux sont un élément essentiel de la gestion du trafic urbain, contribuant à la sécurité routière et à la fluidité du trafic.

## 1.6 Système de gestion des feux tricolores

Il existe des systèmes de gestion des feux tricolores pour surveiller et réguler le trafic aux intersections, garantissant ainsi la fluidité des déplacements des véhicules sur les routes et assurant la sécurité en coordonnant l'utilisation d'un même espace entre des flux de circulation opposés. Il facilite également la gestion du flux de la demande [14].

### 1.6.1 Fonctionnement cyclique des feux de signalisation

1. **Phase d'un feu** Une phase d'un feu correspond à une période pendant laquelle un ou plusieurs flux cohérents sont autorisés à traverser l'intersection. Le cycle d'un feu représente la durée entre deux phases identiques de l'intersection et est défini par une séquence de phases. Par exemple, une illustration dans [Fig. 1.7] présente un cycle composé de quatre phases [10].
2. **Cycle d'un feu** La durée d'un cycle de feu est généralement proportionnelle au nombre de phases qu'il contient. Le temps de dégagement, également connu sous le nom de rouge intégral, est le temps nécessaire pour que tous les véhicules quittent l'intersection avant le passage à une autre phase. Une augmentation de la durée du cycle peut accroître la capacité de l'intersection en réduisant la part du temps consacré au dégagement [10].

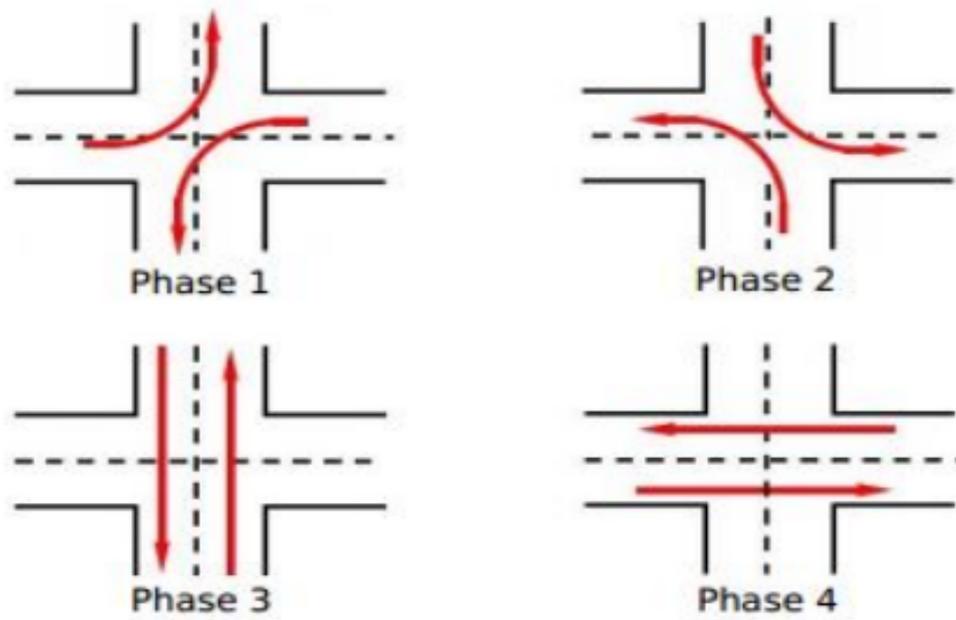


FIGURE 1.7 – Plan d'un cycle composé de quatre phases [44]

### 1.6.2 L'évolution des systèmes de gestion des feux tricolores

Au fil des décennies, la gestion des feux de circulation a évolué pour répondre aux besoins croissants de la circulation routière. Les principaux facteurs influençant cette évolution sont l'augmentation du nombre de véhicules sur les routes, les avancées technologiques et la volonté de promouvoir une mobilité durable [15].

Voici un résumé des différentes phases de cette évolution :

Phase	Période	Description
Phase 1	1868 - 1920	Les premiers feux de circulation alimentés au gaz étaient gérés manuellement par des agents de police. Malheureusement, les premiers essais au Royaume-Uni ont échoué.
Phase 2	1920 - 1980	Le développement des plans à temps fixe a débuté pour répondre à la croissance de la congestion urbaine et établir des objectifs de base pour la gestion des feux.
Phase 3	Années 1970 à aujourd'hui	Les efforts pour résoudre les problèmes de congestion ont conduit à l'introduction de systèmes de signalisation activés par les véhicules pour améliorer l'efficacité.
Phase 4	Depuis 1997	Les systèmes de gestion des feux de circulation sont devenus plus avancés, intégrant des technologies de pointe et des agents intelligents pour réduire la congestion.

### 1.6.3 La signalisation intelligente

La signalisation intelligente est un système de gestion du trafic routier qui utilise des données en temps réel pour ajuster les cycles de feux tricolores aux intersections. Ce système analyse continuellement le flux de circulation à l'aide de capteurs et d'autres technologies pour déterminer la meilleure configuration des feux en fonction des conditions actuelles de la route [52].

L'objectif principal de la signalisation intelligente est d'optimiser la circulation routière en réduisant les temps d'attente, en minimisant les congestions et en améliorant la sécurité des usagers de la route.

## 1.7 Architecture des feux de signalisation intelligentes

Pour améliorer la fluidité du trafic et réduire les temps d'attente des véhicules aux intersections, il est essentiel de mettre en place une architecture générale de signalisation intelligente du trafic. Cette architecture doit comprendre deux modules clés, chacun jouant un rôle essentiel dans l'optimisation de la circulation routière : Module de collecte des statistiques de trafic et Module de contrôle du trafic.

### 1.7.1 Module de collecte des statistiques de trafic

Ce premier module est chargé de recueillir des données sur l'activité routière en temps réel aux intersections. Il utilise une variété de technologies de détection, telles que des capteurs de boucles électromagnétiques, des caméras intelligentes, des réseaux de capteurs sans fil, des radars à hyperfréquences et des fibres optiques, pour recueillir des informations précises sur le nombre de véhicules, la vitesse, les mouvements et d'autres paramètres de trafic pertinents [12].

Les différentes technologies utilisées dans le cadre du module de collecte des statistiques de trafic pour recueillir des informations sur l'activité courante du trafic sur les chemins [53] sont :

1. **Boucles électromagnétiques** : Ces boucles sont enterrées dans la chaussée et détectent la présence de véhicules en mesurant les changements dans le champ électromagnétique lorsqu'un véhicule passe au-dessus. Elles sont largement utilisées dans les systèmes de contrôle de la circulation routière.
2. **Capteurs vidéo (caméra intelligente)** : Les caméras intelligentes capturent des images en temps réel de la circulation routière et utilisent des algorithmes de traitement d'image pour détecter et suivre les véhicules. Elles peuvent fournir des informations détaillées sur le trafic, telles que le nombre de véhicules, les mouvements, et les temps d'attente.
3. **Réseaux de capteurs sans fil** : Ces réseaux utilisent des capteurs sans fil répartis le long des routes pour surveiller le trafic. Ils peuvent collecter une variété de données, y

compris le flux de véhicules, la vitesse et les conditions météorologiques, et transmettre ces informations à un centre de contrôle.

4. **Capteurs à hyperfréquence (radar) :** Les capteurs à hyperfréquence, tels que les radars, émettent des ondes électromagnétiques à haute fréquence et mesurent le temps nécessaire pour que ces ondes rebondissent sur les véhicules. Ils peuvent fournir des informations sur la vitesse, la taille et la position des véhicules.
5. **Fibres optiques :** Les fibres optiques sont utilisées pour surveiller le trafic en détectant les changements dans la lumière réfléchiée causés par les véhicules. Elles peuvent être intégrées dans le revêtement de la route ou le long des bordures de la chaussée pour une surveillance discrète et précise du trafic.

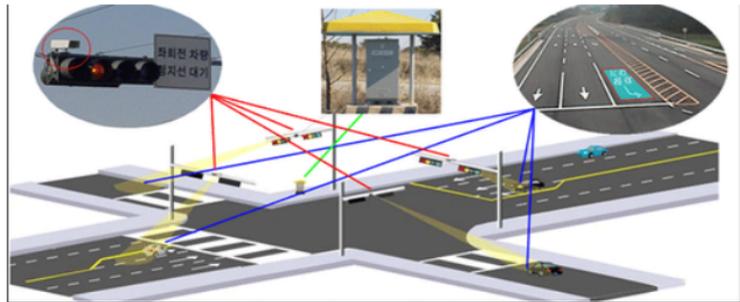


FIGURE 1.8 – Les différents capteurs [45]

### 1.7.2 Module de contrôle du trafic

Le deuxième module est chargé d’analyser les données recueillies par le module précédent et de réguler le fonctionnement des feux de signalisation en conséquence. En utilisant des algorithmes intelligents et des stratégies de contrôle adaptatives, ce module ajuste les cycles de feux tricolores aux intersections pour optimiser la circulation, minimiser les temps d’attente des véhicules et réduire les congestions routières [1].

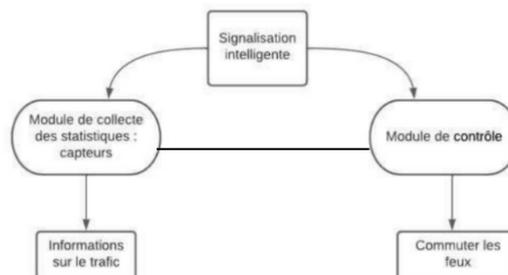


FIGURE 1.9 – Structure de la signalisation intelligente.

## 1.8 Logiciels de gestion et de commande des feux de signalisation

Les outils spécialisés dans la gestion du trafic routier et la commande des feux de signalisation :

1. **TRANSYT (Traffic Network Study Tool)** : Il s'agit d'un logiciel d'optimisation des feux de signalisation à temps fixe, développé initialement par le TRRL au Royaume-Uni en 1996. TRANSYT a été continuellement amélioré, avec sa dernière version, la version 10 new release, publiée en 1996. Ce système est largement utilisé comme référence pour évaluer l'efficacité des systèmes de régulation en temps réel [8].

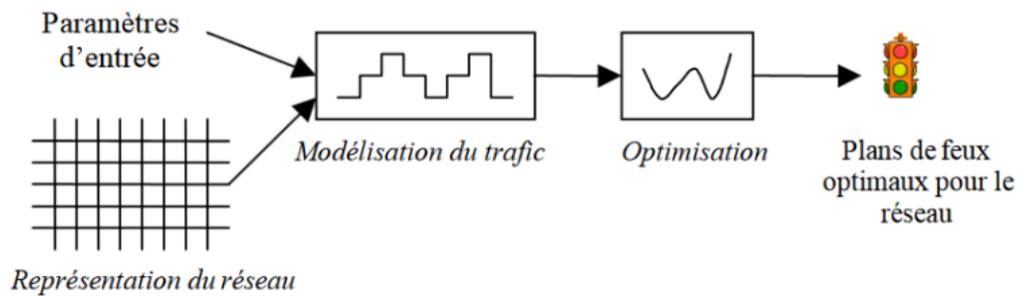


FIGURE 1.10 – L'architecture de TRANSYT

2. **SCOOT (Split Cycle and Offset Optimization Technique)** : SCOOT est un système entièrement adaptatif qui utilise des données collectées par des détecteurs routiers pour calculer les paramètres permettant de réduire les retards et les arrêts aux intersections. Il a été testé et évalué dans diverses situations réelles grâce à la technique d'enquête sur les véhicules en déplacement [9].

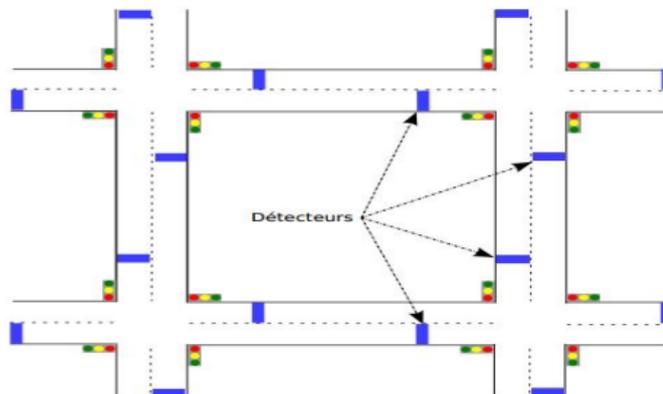


FIGURE 1.11 – SCOOT

3. **SCATS (Sydney Co-ordinated Adaptive Traffic System)** : SCATS est un système partiellement décentralisé et adaptatif qui utilise des données de capteurs routiers pour ajuster la signalisation en fonction de la situation du trafic. Il est largement déployé dans plusieurs villes du monde, notamment en Australie, et se distingue par son approche hiérarchique et son utilisation de bibliothèques de durées de feu vert et de décalages [44].
4. **PRODYN (Programmation DYNamique)** : Ce système, développé par le CERT au début des années 1980, a été testé dans la Zone Expérimentale et Laboratoires de Trafic de Toulouse. Il est utilisé dans plusieurs villes et est industrialisé par les sociétés Garbarini et GTMH. PRODYN se distingue par son approche de programmation dynamique pour la gestion du trafic [8].

## Conclusion

En conclusion, l'infrastructure routière constitue le fondement essentiel de nos déplacements quotidiens, de la simple route locale aux autoroutes interurbaines. Nous avons examiné divers aspects de cette infrastructure, y compris le trafic routier, les réseaux routiers et les intersections complexes, ainsi que les défis liés à leur gestion efficace. La gestion du trafic routier est cruciale pour assurer la sécurité des usagers de la route, réduire la congestion et préserver l'environnement, nécessitant l'utilisation de diverses méthodes et technologies de gestion du trafic.

## **Chapitre 2**

# **OPTIMISATION DU CONTRÔLE DES FEUX DE SIGNALISATION : APPROCHES MATHÉMATIQUES**

### **Introduction**

Chaque année, le nombre de conducteurs sur nos routes augmente, dépassant souvent la capacité des infrastructures, ce qui crée des embouteillages et des files d'attente interminables. Pour résoudre ce problème, de nouvelles méthodes de contrôle du trafic ont émergé, parmi lesquelles les feux de signalisation intelligents se démarquent.

Dans ce chapitre, nous aborderons deux principaux sujets. D'abord, nous explorerons les avancées dans le domaine des feux de signalisation intelligents et passerons en revue les recherches pertinentes. Ensuite, nous présenterons une nouvelle méthode de contrôle des feux de circulation, basée sur un modèle de jeu non coopératif avec contraintes, développé dans [30]. Ce modèle, issu de ses travaux de thèse, prend en compte les interactions stratégiques des mouvements à une intersection. Il s'appuie sur des données en temps réel collectées via des communications entre véhicules et infrastructures, permettant d'ajuster les contrôles des feux en fonction de l'équilibre de Nash généralisé.

### **2.1 Exploration des Recherches Précédentes**

Les recherches antérieures se concentrent principalement sur les approches mathématiques et leurs diverses techniques.

### **2.1.1 La Programmation Dynamique (PD)**

La Programmation Dynamique (PD) est une méthode mathématique essentielle dans le domaine du contrôle des feux de circulation. Elle décompose le problème complexe en une série de sous-problèmes gérables, en segmentant le temps en intervalles discrets. La PD utilise des algorithmes de recherche pour trouver la solution optimale à chaque sous-problème, en ajustant les paramètres de contrôle tels que la durée des feux de signalisation. De nombreux systèmes de contrôle adaptatif des feux de circulation se sont entièrement appuyés sur la méthode PD [48,49], permettant ainsi d'optimiser la synchronisation des feux en temps réel et de réduire les temps d'attente des véhicules, contribuant ainsi à améliorer la fluidité du trafic et à réduire les temps de trajet.

Les auteurs dans [16, 51] ont proposé une approche novatrice pour le contrôle des feux de circulation, combinant la Programmation Dynamique (PD) avec des réseaux de neurones. Cette méthode hybride utilise la PD pour calculer les temps optimaux des feux verts pour une intersection isolée, en supposant que le schéma d'arrivée des véhicules est connu à l'avance. Les réseaux de neurones sont ensuite utilisés pour ajuster dynamiquement les temps des feux verts en fonction des conditions de trafic réelles, en apprenant à partir de données historiques et en temps réel. Cette approche permet d'améliorer l'efficacité du contrôle des feux de circulation en réduisant les temps d'attente, en optimisant le flux de trafic et en améliorant la sécurité routière.

### **2.1.2 Les heuristiques et les métaheuristiques**

Les heuristiques et les métaheuristiques [13, 18] sont des approches efficaces pour le contrôle des feux de signalisation, permettant de trouver des solutions dans des environnements complexes. Les heuristiques utilisent des règles empiriques pour guider la recherche de solutions acceptables, tandis que les métaheuristiques sont des techniques plus générales qui explorent efficacement l'espace de recherche pour trouver des solutions de haute qualité. Ces méthodes sont utilisées pour concevoir des stratégies de contrôle adaptatives qui optimisent la fluidité du trafic et la sécurité routière en temps réel.

### **2.1.3 La théorie des jeux**

La théorie des jeux offre une approche sophistiquée pour concevoir des stratégies de contrôle des feux de signalisation, en permettant de modéliser mathématiquement les interactions entre les mouvements des véhicules et la répartition du temps alloué aux feux. Plusieurs études explorent diverses formes de jeux pour représenter les scénarios de coopération ou de conflit entre les différentes approches d'une intersection.

Dans [23] les auteurs explorent l'application de la théorie des jeux à la prise de décision

en temps réel pour le contrôle des feux de signalisation basé sur l'Internet des objets (IoT). Les auteurs examinent comment cette approche peut améliorer la gestion du trafic urbain en optimisant la fluidité du trafic, en réduisant les temps d'attente et en améliorant la sécurité routière.

Dans [42], les auteurs ont présenté une étude sur le contrôle des feux de signalisation à une intersection unique, basée sur un modèle de jeu de coopération à deux joueurs. Les auteurs explorent comment ce modèle peut être utilisé pour améliorer la coordination entre les différentes phases des feux de signalisation à l'intersection, afin d'optimiser la fluidité du trafic et de réduire les temps d'attente.

## 2.2 La théorie des jeux et l'optimisation linéaire

La théorie des jeux et l'optimisation linéaire sont deux domaines importants des mathématiques appliquées, chacun ayant ses propres concepts, méthodes et applications distinctes.

### 2.2.1 La programmation linéaire (PL)

Le PL est une méthode d'optimisation mathématique qui vise à trouver la meilleure solution à un problème, étant donné un ensemble de contraintes linéaires et une fonction objectif linéaire à optimiser. Elle est largement utilisée dans divers domaines tels que l'économie, la logistique, la gestion des ressources et l'ingénierie...

Les problèmes de programmation linéaire sont souvent formulés comme suit [6] :

1. **Fonction objectif** : Maximiser ou minimiser une fonction linéaire de certaines variables de décision.
2. **Contraintes** : Les variables de décision sont soumises à des contraintes linéaires.
3. **Variables de décision** : Ce sont les quantités dont nous cherchons à déterminer les valeurs optimales.

Voici la formulation générale d'un problème de programmation linéaire :

**Maximiser (ou Minimiser) :**

$$c_1x_1 + c_2x_2 + \dots + c_nx_n \tag{2.1}$$

**Sous les contraintes :**

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m \\ x_1, x_2, \dots, x_n \geq 0 \end{array} \right.$$

Où :

- $x_1, x_2, \dots, x_n$  : sont les variables de décision.
- $c_1, c_2, \dots, c_n$  : sont les coefficients de la fonction objectif.
- $a_{ij}$  : sont les coefficients des contraintes.
- $b_1, b_2, \dots, b_m$  : sont les termes constants des contraintes.
- $m$  : est le nombre de contraintes .
- $n$  : est le nombre de variables de décision.

### 2.2.2 La théorie des jeux

La TJ est une branche des mathématiques qui étudie les interactions stratégiques entre des agents rationnels. Elle fournit des outils pour analyser les décisions que prennent ces agents en situation de conflit ou de coopération, où le gain ou la perte de chaque participant dépend non seulement de ses propres actions mais aussi de celles des autres.

### 2.2.3 Les Concepts Clés de la TJ

- (a) **Un jeu** Un jeu représente une interaction stratégique entre deux joueurs ou plus, où chaque participant prend des décisions qui influencent le déroulement du jeu. Les éléments définissant un jeu incluent les joueurs, les stratégies et les récompenses associées à chaque issue ou profil d'action.
- (b) **Un joueur** Dans la théorie des jeux, un joueur est une entité, comme une personne ou une organisation, qui prend des décisions dans une situation où le résultat dépend des actions de tous les participants. Les joueurs choisissent des stratégies pour maximiser leurs gains potentiels tout en tenant compte des actions des autres joueurs.
- (c) **Stratégie** : une stratégie est un plan d'action complet qu'un joueur peut suivre. Ce plan d'action détermine les choix que le joueur fera dans chaque situation possible qu'il pourrait rencontrer au cours du jeu [4] .

**Types de Stratégies :**

- **Stratégie pure d'un joueur  $i$**  : Une stratégie pure est une action spécifique, ou une combinaison d'actions, que le joueur  $i$  choisit de manière déterminée et sans incertitude. Il n'y a pas de hasard ou de probabilités impliquées dans ce choix.

- **Stratégie mixte** : Une stratégie mixte est une approche où le joueur sélectionne de manière aléatoire parmi plusieurs actions possibles, selon une distribution de probabilités. Cela signifie que chaque action pure a une certaine probabilité d'être choisie. Elle est représentée par un vecteur  $\alpha = (\alpha_1, \dots, \alpha_{n_i}) \in \Delta_i$ , où  $\Delta_i$  est l'ensemble des stratégies mixtes du joueur  $i$  :

$$\Delta_i = \left\{ \alpha = (\alpha_1, \dots, \alpha_{n_i}) \in \mathbb{R}^{n_i} : 0 \leq \alpha_j \leq 1, \forall j \in \{1, \dots, n_i\}, \sum_{j=1}^{n_i} \alpha_j = 1 \right\}$$

où  $\alpha_j$  est la probabilité que le joueur joue sa  $j$ -ème stratégie pure,  $n_i$  est le nombre de stratégies pures du joueur, et  $\Delta_i$  représente l'ensemble des stratégies mixtes du joueur  $i$ .

(d) **Gain** :

Le gain d'un joueur, également désigné par les termes "paiement" ou "utilité", se présente sous forme d'une valeur numérique qui traduit le résultat ou la récompense obtenue par ce joueur à l'issue d'un jeu. Ce gain est déterminé en fonction des actions qu'il a sélectionnées ainsi que celles choisies par les autres participants.

### 2.2.4 Présentation d'un jeu sous forme normale

Un jeu sous forme normale est caractérisé par un triplet :

$$\langle I, \{X_i\}_{i \in I}, \{f_i\}_{i \in I} \rangle \tag{2.3}$$

Où :

- $I = \{1, 2, \dots, n\}$  : représente l'ensemble des joueurs.
- $X_i$  : désigne l'ensemble des stratégies accessibles pour chaque joueur  $i$ . Pour une issue  $x$  du jeu, un vecteur  $x = (x_1, x_2, \dots, x_n)$  est défini, où  $x_i \in X_i$  indique la stratégie choisie par le joueur  $i$ .

- Les gains (ou utilités) pour les joueurs sont exprimés à travers une fonction  $f_i : X_1 \times X_2 \times \dots \times X_n \rightarrow \mathbb{R}$ , définie sur l'ensemble des issues  $x \in X = \prod_{i=1}^n X_i$  [37] .

### 2.2.5 Méthodes de Résolution en Théorie des Jeux

En théorie des jeux, les méthodes de résolution sont des outils utilisés pour identifier les équilibres d'un jeu. Ces équilibres servent de prédictions sur le déroulement du jeu et suggèrent

les résultats probables.

### 1. Stratégie de meilleure réponse :

Une stratégie de meilleure réponse est une stratégie qu'un joueur choisit pour maximiser son gain, étant donné les actions des autres joueurs. En d'autres termes, c'est la stratégie optimale qu'un joueur adopte en réponse aux stratégies choisies par les autres participants. Une stratégie de meilleure réponse est souvent utilisée dans le contexte de jeux à somme non nulle, où les joueurs cherchent à maximiser leur propre utilité en réagissant de manière adaptée aux actions des autres joueurs [37] .

Une stratégie pure  $x_i$  est considérée comme une meilleure réponse aux stratégies des autres joueurs  $x_{-i}$  dans le jeu (2.3) si :

$$f_i(x_i, x_{-i}^*) \geq f_i(y_i, x_{-i}^*); \quad \forall y_i \in X_i. \quad (2.4)$$

$$X_{-i} = \prod_{I \setminus \{i\}} X_j = X_1 \times \dots \times X_{i-1} \times X_{i+1} \times \dots \times X_N$$

On note  $B(x_{-i})$  l'ensemble de toutes les stratégies qui constituent une meilleure réponse à  $x_{-i}$  :

$$B(x_{-i}^*) = \{x_i \in X_i, f_i(x_i, x_{-i}^*) \geq f_i(y_i, x_{-i}^*); \quad \forall y_i \in X_i, \quad \forall i \in I\} \quad (2.5)$$

### 2. Équilibre de Nash :

Un équilibre de Nash, introduit par John Nash, est une situation dans un jeu où aucun joueur n'a intérêt à changer sa stratégie, étant donné les stratégies des autres joueurs. C'est un concept fondamental en théorie des jeux, utilisé pour prédire le comportement des acteurs dans des situations stratégiques [37] .

Une situation  $x^* = (x_1^*, x_2^*, \dots, x_N^*)$  dans le jeu (2.3) constitue un équilibre de Nash en stratégies pures si, pour chaque joueur :

$$f_i(x_i^*, x_{-i}^*) > f_i(x_i, x_{-i}^*); \quad \forall x_i \in X_i, \quad \forall i \in I \quad (2.6)$$

On dit qu'une stratégie  $x_i \in X_i$  du joueur  $i$  est dominée dans un jeu s'il existe une autre stratégie  $y_i \in X_i$  telle que :

$$f_i(y_i, x_{-i}) \geq f_i(x_i, x_{-i}); \quad \forall x_{-i} \in X_{-i} \quad (2.7)$$

où :

$$(y_i, x_{-i}) = (x_1, \dots, x_{i-1}, y_i, x_{i+1}, \dots, x_N)$$

## 2.3 Problématique

Le problème de gestion du trafic routier dans les zones urbaines est devenu une préoccupation majeure en raison de la croissance constante du nombre de véhicules. Les intersections régulées par des feux de signalisation sont des points critiques où des embouteillages peuvent se former, entraînant des retards, une augmentation de la consommation de carburant et une pollution accrue. Une gestion efficace des feux de signalisation est donc essentielle pour améliorer la fluidité du trafic et réduire les impacts négatifs associés.

La régulation des feux de signalisation à une intersection implique de diviser le temps en cycles et de déterminer la durée optimale des feux verts pour chaque phase de circulation. Traditionnellement, ce problème est traité par des modèles de programmation dynamique, des modèles de la théorie des jeux ou des modèles stochastiques. Cependant, ces approches peuvent ne pas toujours garantir une optimisation globale du système, surtout dans des conditions de trafic dynamique et complexe.

Dans ce mémoire, nous nous intéressons aux modèles de la théorie des jeux qui visent à optimiser le passage sécurisé et rapide de tous les usagers du réseau routier arrivant à une intersection par différentes routes. Pour ce faire, nous commençons par regrouper tous les axes routiers débouchant sur une intersection en une seule phase, permettant ainsi aux usagers de traverser en même temps en toute sécurité. Chaque phase de circulation est considérée comme un "joueur" cherchant à minimiser son temps d'attente tout en respectant certaines contraintes de durée. Ce modèle, développé dans [30], il a d'abord été appliqué à une intersection à deux phases. Nous avons ensuite étendu son application à une intersection à quatre phases pour évaluer sa performance et sa robustesse dans des situations plus complexes.

## 2.4 Présentation d'une intersection

Dans de nombreux travaux sur la régulation des feux de signalisation, les mouvements distincts d'une intersection sont structurés en phases et le temps est fractionné en cycles. Chaque période de cycle correspond à une période de contrôle, et le défi est de déterminer la durée du feu vert pour chaque phase de l'intersection au début de chaque cycle. Il est donc possible de représenter le réseau d'intersections en un graphe orienté, où chaque sommet est une intersection et chaque arc est un mouvement d'une intersection à une autre.

Le contrôle des feux de signalisation nécessite une gestion efficace des mouvements à une intersection. Pour ce faire, ces mouvements sont organisés en phases, et le temps est divisé en cycles. Chaque cycle correspond à une période où le contrôle est actif, et l'objectif est de déterminer la durée du feu vert pour chaque phase au début de chaque cycle.

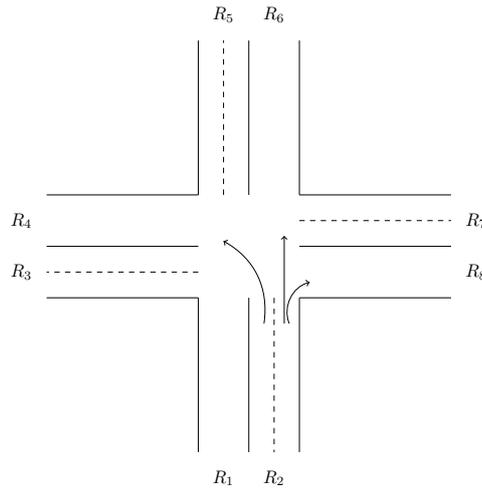


FIGURE 2.1 – Exemple d’une intersection  $T$ .

Les intersections sont représentées par des sommets dans un graphe orienté, où les arcs décrivent les mouvements entre les intersections. Chaque intersection possède des approches entrantes et sortantes, chacune étant subdivisée en voies pour différents types de mouvements. Les mouvements possibles à une intersection sont regroupés dans un ensemble  $M_T$  qui est donné, pour l'exemple de la Figure (2.1) par :

$$\{(R_5, R_4), (R_5, R_1), (R_2, R_6), (R_2, R_8), (R_2, R_4), (R_5, R_8), (R_3, R_8), (R_3, R_1), (R_3, R_6), (R_7, R_4), (R_7, R_6), (R_7, R_1)\}$$

Chaque intersection est définie par un ensemble de sommets  $S$  et un ensemble d’arcs  $L$ .

Les mouvements à chaque intersection sont regroupés en phases, notées  $P_T$ . Le nombre de phases dépend de la distribution du trafic et de la configuration de l’intersection. Chaque phase, représentée par  $P_T = (P_1, P_2, \dots, P_{P_S})$ , décrit les mouvements entre les approches de l’intersection. Les différents mouvements en 4 phases dans la Figure 2.2 :

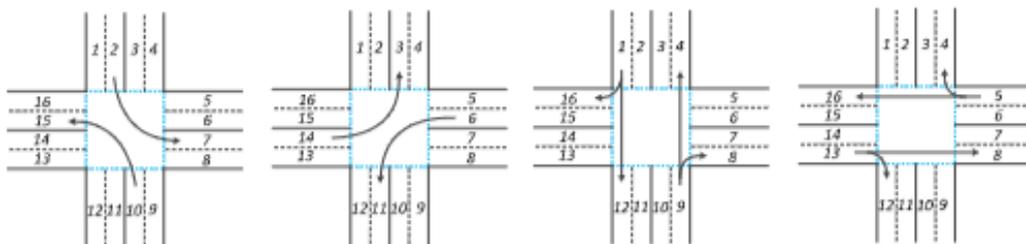


FIGURE 2.2 – Phases typiques d’une intersection [3]

avec :

$$P_1 = \{(13, 8), (13, 12), (5, 16), (5, 4)\}$$

$$P_2 = \{(1, 12), (1, 16), (9, 8), (9, 4)\}$$

$$P_3 = \{(14, 3), (6, 11)\}$$

$$P_4 = \{(2, 7), (10, 15)\}$$

Nous supposons que le contrôleur de chaque intersection et les véhicules peuvent échanger des informations entre eux. En se basant sur les données recueillies, le contrôleur évalue la demande de trafic à l'intersection. Cette demande est définie par trois variables [40] :

- $N_i$  : le nombre de véhicules présents dans la phase  $P_i$  au début du cycle .
- $S_i$  : le débit de sortie des véhicules de la phase  $P_i$  pendant le cycle .
- $E_i$  : le débit d'arrivée des véhicules dans la phase  $P_i$  pendant le cycle.

## 2.5 Modélisation du Trafic

Dans cette partie, nous allons introduire un jeu non coopératif avec contraintes pour représenter le problème de régulation des feux de signalisation à une intersection.

1. **Ensemble des joueurs** : Chaque phase  $P_i$  est sous le contrôle d'un joueur  $i$ , où l'ensemble des joueurs du jeu est défini comme :

$$I = \{1, 2, 3, 4\}$$

2. **Définition des Stratégies** : Chaque joueur  $i$  doit choisir une durée  $g_i$  de feu vert pour sa phase, ne dépassant pas la durée totale du cycle,  $C$ . L'ensemble des stratégies possibles pour un joueur  $i$  est :

$$X_i = [0, C], \quad i \in I$$

Une fois que chaque joueur  $i \in I$  ait choisi sa stratégie  $g_i \in X_i$ , on obtient une issue du jeu

$$g = (g_1, \dots, g_4) \in X = \prod_{i=1}^4 X_i.$$

Pour mettre en évidence le choix d'un joueur  $i \in I$ , nous utiliserons la notation

$$(g_i, \dots, g_{-i}^*) = (g_1^*, \dots, g_{i-1}^*, g_i, g_{i+1}^*, \dots, g_4^*)$$

où :

$$g_i \in X_i; g_{-i}^* \in \prod_{j=1, j \neq i}^I X_j$$

3. **Contraintes de Durée** : Les décisions des joueurs doivent respecter les contraintes suivantes [40] :

(a) **Contrainte 1 : Durée Minimale du Feu Vert**

Pour permettre à un véhicule de passer l'intersection, la durée  $g_i$  doit être supérieure ou égale à une durée minimale  $g_{\min, i}$  :

$$g_i \geq g_{\min, i} \quad \text{ou} \quad h_{i1}(g_i, g_{-i}) = -g_i + g_{\min, i} \leq 0, \quad \forall i \in I. \quad (2.8)$$

**(b) Contrainte 2 : Limitation de la Durée du Feu Vert**

La durée  $g_i$  ne doit pas dépasser le temps nécessaire pour écouler les véhicules en file d'attente, afin d'éviter un gaspillage de temps pour les autres phases :

$$g_i \leq \frac{N_i + E_i \sum_{j=1, j \neq i}^{PJ=4} g_j}{S_i - E_i} \quad \text{ou} \quad h_{i2}(g_i, g_{-i}) = (S_i - E_i)g_i - E_i \sum_{j=1, j \neq i}^{PJ} g_j - N_i \leq 0, \quad \forall i \in I \quad (2.9)$$

**(c) Contrainte 3 : Somme des Durées**

La somme des durées des feux verts de toutes les phases ne doit pas dépasser la durée totale du cycle  $C$  :

$$g_i \leq C - \sum_{j \in I \setminus i} g_j \quad \text{ou} \quad h_{i3}(g_i, g_{-i}) = \sum_{i=1}^{P_T} g_i - C \leq 0, \quad \forall i \in I \quad (2.10)$$

Une issue  $g = (g_i, g_{-i}) \in X$  est réalisable si :

$$h_{ik}(g_i, g_{-i}) \leq 0, \forall k \in \{1, 2, 3\}, \forall i \in I \quad (2.11)$$

En termes de théorie des méta-jeux, une issue du méta-jeu est un vecteur  $g = (g_1, g_2, g_3, g_4) \in R^4$  vérifiant les contraintes (2.11) ou ce qui est équivalent :

$$g_i \in D_i(g_{-i}), \forall i \in I$$

où la règle de décision  $D_i(\cdot) : X_{-i} \rightarrow 2^{X_i}$  est définie par :

$$D_i(g_{-i}) = \{g_i \in X_i, h_{ik}(g_i, g_{-i}) \leq 0, k \in \{1, 2, 3\}\} \quad (2.12)$$

**4. Objectif des Joueurs** L'objectif de chaque joueur  $i \in I$  est de minimiser le nombre de véhicules sur les liens de sa phase  $P_i$  à la fin d'un cycle, qui est donné par :

$$\Pi_i(g_i, g_{-i}) = N_i + E_i \sum_{j=1, j \neq i}^{P_T} g_j + (E_i - S_i)g_i \quad (2.13)$$

où :

- $N_i$  : le nombre de véhicules présents dans la phase  $P_i$  au début du cycle .
- $S_i$  : le débit de sortie des véhicules de la phase  $P_i$  pendant le cycle .
- $E_i$  : le débit d'arrivée des véhicules dans la phase  $P_i$  pendant le cycle.

**5. La forme normale du jeu** Le problème de contrôle des feux de signalisation est ainsi modélisé comme un méta-jeu (jeu non coopératif avec contraintes) défini par [20] :

$$\Gamma = \langle I, \{X_i\}_{i \in I}, \{D_i\}_{i \in I}, \{\Pi_i\}_{i \in I} \rangle \quad (2.14)$$

## 6. Équilibre de Nash Généralisé pour le Méta-jeu

Le méta-jeu (2.14) utilise le concept d'équilibre de Nash généralisé (ENG), également appelé équilibre social, comme solution.

- (a) **Définition du GNE** : Un vecteur  $g^* \in X$  est un GNE du méta-jeu (2.14) s'il satisfait les conditions suivantes :

$$g_i^* \in D_i(g_{-i}^*) \quad \forall i \in I \quad (2.15)$$

$$\Pi_i(g_i^*, g_{-i}^*) \leq \Pi_i(g_i, g_{-i}^*), \quad \forall g_i \in D_i(g_{-i}^*), \quad \forall i \in I \quad (2.16)$$

- (b) **Conditions d'équilibre** :

La condition (2.15) garantit que  $g^*$  est une issue réalisable du jeu (2.14). La condition (2.16) assure que toute déviation unilatérale d'un joueur vers une autre stratégie ne lui procurerait pas une meilleure utilité que celle obtenue avec  $g^*$ .

- (c) **Issue réalisable** : Une issue  $g = (g_i, g_{-i})$  du jeu (2.16) est réalisable si elle satisfait  $g \in S$ , où :

$$S = \{g \in X \mid h(g) \leq 0\} \quad (2.17)$$

avec :

$$h(g) = (h_{ik}(g_i, g_{-i}); k \in \{1, 2, 3\}, i \in I)$$

est une fonction vectorielle représentant les contraintes du jeu.

- (d) **Propriétés de S** : L'ensemble  $S$  des issues réalisables est non vide, convexe, et compact, grâce à la linéarité des contraintes  $h_{ik}(\cdot)$  et la convexité des ensembles  $X_i$ .

- (e) **Optimisation pour GNE** : Un vecteur  $g^* \in S$  est un GNE s'il minimise :

$$\min_{z_i \in X_i / (z_i, g_{-i}^*) \in S} \Pi_i(z_i, g_{-i}^*), \quad \forall i \in I \quad (2.18)$$

## 7. Conditions d'Existence d'un Équilibre de Nash Généralisé

Pour établir les conditions d'existence d'un équilibre de Nash généralisé (GNE) pour le jeu (2.14), nous suivons la méthode présentée dans [17]. Le jeu est associé à la fonction  $P(g, z) = \sum_{i \in I} \Pi_i(z_i, g_{-i})$ , définie sur l'ensemble des stratégies réalisables  $S \times S$ .

La fonction  $P(g, z)$  est linéaire par rapport aux variables  $g$  et  $z$ . Cette linéarité nous permet d'appliquer le Théorème 1 de [17] et le Théorème du point fixe de Kakutani pour démontrer l'existence d'un GNE pour le jeu (2.14).

**Théorème 1.** *Le méta-jeu (2.14) [30] possède un équilibre de Nash généralisé.*

Après avoir confirmé l'existence d'un GNE pour le méta-jeu (2.14), qui garantit une répartition équilibrée des durées de feu vert pour les différentes phases d'une intersection (T),

nous aborderons dans la prochaine section le processus de calcul de cette solution d'équilibre généralisé.

## 2.6 Résolution

L'équilibre généralisé de Nash pour le méta-jeu (2.14) est déterminé en utilisant l'approche par les fonctions de meilleures réponses  $MR_i$ , où chaque joueur  $i$  résout un programme linéaire donné par :

$$MR_i(x_{-i}) = \arg \min_{g_i} \Pi_i(g_i, g_{-i}) \text{ s.t. } g_i \in D_i(g_{-i}), \forall i \in I. \quad (2.19)$$

### Formulation du Problème (2.19) sous forme d'un problème Linéaire d'Équilibre Généralisé (LGNEP)

Le problème est connu sous le nom de Problème Linéaire d'Équilibre de Nash Généralisé (LGNEP) et est défini par [30] :

$$\min_{g_i} \Pi_i(g_i, g_{-i}) \text{ s.t. } g_i \in D_i(g_{-i}), \forall i \in I. \quad (2.20)$$

où la fonction objectif  $\theta_i$  est donnée par :

$$\Pi_i(g_i, g_{-i}) = (E_i - S_i)g_i, \forall i \in I. \quad (2.21)$$

**Formulation Matricielle des Contraintes :** Les contraintes (2.9) (2.10) et (2.11) sont exprimées de manière matricielle :

$$D_i(g_{-i}) = \{g_i \in X \mid A_{ii}g_i + \sum_{j \in I, j \neq i} A_{ij}g_j \leq b_i\}, \forall i \in I, \quad (2.22)$$

où les matrices  $A$  et le vecteur  $b$  sont définis respectivement par :

$$b_i = \begin{pmatrix} -g_{\min, i} \\ N_i \\ C \end{pmatrix}, A_{ii} = \begin{pmatrix} -1 \\ S_i - E_i \\ 1 \end{pmatrix}, A_{ij} = \begin{pmatrix} 0 \\ -E_i \\ 1 \end{pmatrix}. \quad (2.23)$$

Ainsi, la recherche d'un équilibre généralisé de Nash du jeu (2.14) se ramène à la résolution d'un problème d'équilibre de Nash généralisé avec contraintes partagées (LGNEPSC) :

Sous forme matricielle, le LGNEP devient [30] :

$$\min_{g_i} \Pi_i(g_i, g_{-i}) \text{ s.t. } \tilde{A}(g_i, g_{-i}) \leq \tilde{b}, \forall i \in I \quad (2.24)$$

avec :

$$A = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ S_1 - E_1 & -E_1 & -E_1 & -E_1 \\ -E_2 & S_2 - E_2 & -E_2 & -E_2 \\ -E_3 & -E_3 & S_3 - E_3 & -E_3 \\ -E_4 & -E_4 & -E_4 & S_4 - E_4 \end{bmatrix} \quad (2.25)$$

et

$$b = \begin{bmatrix} -g_{\min,1} \\ -g_{\min,2} \\ -g_{\min,3} \\ -g_{\min,4} \\ N_1 \\ N_2 \\ N_3 \\ N_4 \\ C \end{bmatrix} \quad (2.26)$$

**Conditions de Karush-Kuhn-Tucker (KKT) du problème 2.24 :**

La résolution du LGNEPSC 2.24 utilise les conditions KKT données par [30] :

$$\begin{cases} \mu_i - \omega_i + (A_i)^T \lambda_i = 0, \\ Ag^* - b + v_i = 0, \\ v_i \geq 0, \lambda_i \geq 0, v_i^T \lambda_i = 0. \end{cases} \quad (2.27)$$

**Définition et Propriétés des Équilibres de Nash Généralisés Normalisés :** Un équilibre de Nash généralisé normalisé est défini par l'existence de poids positifs  $p_i > 0$  tels que :

$$\lambda_i = \frac{\lambda}{p_i}, \forall i \in I \quad (2.28)$$

où  $\lambda$  est un multiplicateur commun.

**Théorème de Caractérisation des Équilibres de Nash Généralisés Normalisés :**

Une issue  $g^*$  est un équilibre de Nash généralisé normalisé si et seulement si il existe des poids  $p_i^* > 0$  tels que  $g^*$  est une solution du programme linéaire [30] :

$$\min \sum_{i \in I}^{P_T} f_i g_i \text{ s.t. } (g_i, g_{-i}) \in S \quad (2.29)$$

avec :

$$f_i = p_i^*(E_i - S_i), \forall i \in I \quad (2.30)$$

où :  $p_i$  et définis par (2.28).

## Algorithme

La procédure de calcul de cet équilibre utilise la méthode du Simplexe duale, qui est plus efficace en temps d'exécution que la méthode du point intérieur. L'algorithme inclut les étapes principales de l'algorithme 1.

---

### Algorithm 1 Modèle de Méta-Jeu (MetMod)

---

**Require:**  $J, P_J, C, N_i, S_i, E_i, g_{\min,i}, p_i, \forall i \in I$

- 1: Calculer le vecteur de coûts  $f = (f_i, i \in I)$  en utilisant l'équation  $f_i = p_i^*(E_i - S_i)$
- 2: Construire le vecteur  $b$  en utilisant l'équation  $b_i = \begin{pmatrix} -g_{\min,i} \\ N_i \\ C \end{pmatrix}$
- 3: Construire la matrice  $A$  en utilisant l'équation  $A_{ii} = \begin{pmatrix} -1 \\ S_i - E_i \\ 1 \end{pmatrix}$  et  $A_{ij} = \begin{pmatrix} 0 \\ -E_i \\ 1 \end{pmatrix}$
- 4: Résoudre le programme linéaire (2.29) en utilisant la fonction  $\text{linprog}(f, A, b)$

**Ensure:**  $g_i^*, \forall i \in I$

---

## Conclusion

Dans ce chapitre nous avons exploré diverses méthodes mathématiques pour optimiser le contrôle des feux de signalisation, telles que la programmation dynamique, les heuristiques, les métaheuristiques et la théorie des jeux, visant à améliorer la fluidité du trafic et à réduire les temps d'attente en temps réel. De plus, nous avons étudié la modélisation de la régulation des feux de signalisation comme un jeu non coopératif sous contraintes, où un équilibre de Nash généralisé a été identifié comme solution optimale, utilisant des méthodes efficaces comme les conditions KKT et la méthode du Simplexe duale pour sa résolution.

## **Chapitre 3**

# **OPTIMISATION DU CONTRÔLE DES FEUX DE SIGNALISATION : APPROCHES DE L'INTELLIGENCE ARTIFICIELLE**

### **Introduction**

Dans ce chapitre, nous explorons comment l'intelligence artificielle révolutionne la gestion des feux de signalisation. En utilisant des techniques avancées comme le deep learning et l'apprentissage par renforcement, nous pouvons créer des systèmes adaptatifs et intelligents qui répondent en temps réel aux fluctuations du trafic. Cette approche améliore la fluidité et la sécurité routière, ouvrant la voie à des solutions plus efficaces pour les défis du trafic urbain moderne. De plus, nous présentons un système novateur de contrôle adaptatif des feux de signalisation urbains, basé sur le modèle MetMod [30] et l'apprentissage par renforcement. Ce modèle surmonte les défis liés à l'absence de données en temps réel en utilisant des prédictions basées sur des réseaux de neurones, optimisant ainsi la gestion du trafic urbain même dans des conditions où les données de capteurs traditionnels ne sont pas disponibles.

### **3.1 Les techniques découlant de l'intelligence artificielle**

#### **3.1.1 Les algorithmes génétiques**

Dans [35], les auteurs présentent une étude sur l'utilisation d'un algorithme génétique pour contrôler les feux de signalisation dans un système de gestion du trafic, en fonction des conditions de circulation, en recherchant des configurations efficaces qui minimisent les temps d'at-

tente, réduisent la congestion et améliorent la fluidité du trafic.

### **3.1.2 Les réseaux de neurones artificiels (ANN)**

Dans [29] [22] [33] les auteurs explorent l'application des réseaux de neurones artificiels (ANN) dans le domaine de la gestion du trafic routier et de l'éclairage urbain. Ils se concentrent sur l'utilisation de techniques basées sur les ANN pour résoudre des problèmes spécifiques tels que l'optimisation de la synchronisation des feux de signalisation, le contrôle du flux de circulation et l'amélioration de l'efficacité énergétique de l'éclairage des zones urbaines.

### **3.1.3 Systèmes Experts (SE)**

Les SE utilisent des bases de connaissances expertes pour prendre des décisions sur le contrôle des feux de signalisation, en utilisant des règles préétablies basées sur l'expérience humaine et les meilleures pratiques en matière de gestion du trafic.

Les auteurs proposent dans [21] un modèle architectural efficace pour la construction de systèmes experts cognitifs liés à la gestion du trafic dans les villes intelligentes. Il explore les méthodes pour intégrer des systèmes experts avec des technologies cognitives afin de prendre des décisions efficaces en matière de gestion du trafic. Dans [?], les auteurs explorent l'application de la logique floue et des systèmes experts dans le contexte de la gestion dynamique des feux de signalisation, en mettant l'accent sur l'utilisation de technologies telles que les réseaux de capteurs sans fil pour améliorer la réactivité et l'efficacité des systèmes de gestion du trafic urbain.

### **3.1.4 Reinforcement Learning (RL)**

Le RL peut être utilisé pour apprendre à contrôler les feux de signalisation en interagissant avec l'environnement routier et en recevant des récompenses basées sur la performance du système, permettant ainsi d'ajuster les stratégies de contrôle de manière adaptative et dynamique.

Dans [27] les auteurs explorent l'application de l'apprentissage par renforcement profond (deep reinforcement learning) dans le contrôle des feux de signalisation pour la gestion du trafic. Ils présentent différentes approches et modèles de RL pour optimiser les stratégies de contrôle des feux de signalisation, en tenant compte des conditions de circulation variables et de l'impact des données de trafic en temps réel. Les auteurs examinent les avantages et les défis de l'utilisation du RL dans ce contexte, ainsi que les récents progrès réalisés dans le domaine.

Certains auteurs dans [27, 54] se concentrent spécifiquement sur des méthodes telles que

l'apprentissage par renforcement profond, l'apprentissage méta-renforcement et les approches basées sur les valeurs, tandis que d'autres fournissent une vue d'ensemble et une analyse des différents modèles et évaluations utilisés dans la recherche sur le contrôle des feux de signalisation.

Dans [26] les auteurs proposent une technique de contrôle adaptatif des feux de signalisation en utilisant l'apprentissage par renforcement profond. Ils explorent comment cette approche peut permettre aux feux de signalisation de s'adapter dynamiquement aux conditions de circulation changeantes pour optimiser la fluidité du trafic.

### **3.1.5 Algorithmes d'Apprentissage Automatique (ML)**

Les techniques d'apprentissage automatique, telles que les méthodes d'apprentissage supervisé et non supervisé, peuvent être appliquées pour analyser les données historiques et en temps réel, permettant de prédire les schémas de trafic et d'optimiser les temps des feux de signalisation en conséquence.

Dans [34] les auteurs se concentrent sur la prédiction du flux de trafic pour les feux de signalisation intelligents en utilisant des algorithmes d'apprentissage automatique. Ils explorent comment les modèles de machine learning peuvent être utilisés pour anticiper les fluctuations du trafic et ajuster les temps des feux de signalisation en conséquence.

Dans [39] les auteurs se concentrent sur un système de contrôle de feux de signalisation intelligent basé sur l'environnement de trafic en utilisant l'apprentissage profond. Il explorent comment les techniques de deep learning peuvent être appliquées pour détecter et analyser les conditions de circulation en temps réel, afin d'optimiser les temps des feux de signalisation.

Dans [38] les auteurs présentent une analyse comparative des algorithmes de machine learning dans le processus de conception d'un système de contrôle adaptatif des feux de signalisation. Ils examinent les avantages et les limitations de différentes approches de machine learning pour optimiser les temps des feux de signalisation en fonction des conditions de trafic.

Dans [11] les auteurs concentrent sur des intersections isolées et sur la manière dont l'apprentissage par renforcement profond peut être utilisé pour adapter dynamiquement les cycles de feux de signalisation en fonction des conditions de trafic changeantes à ces intersections.

Dans [28] l'auteur propose un réseau d'apprentissage par renforcement profond pour le contrôle du cycle des feux de signalisation. Ils explorent comment cette approche peut être uti-

lisée pour apprendre dynamiquement les stratégies de contrôle des feux de signalisation afin d'optimiser la fluidité du trafic et de réduire la congestion.

### 3.2 Exploration des Paradigmes de l'Intelligence Artificielle : Machine Learning, Deep Learning et Apprentissage par Renforcement

L'exploration des paradigmes de l'intelligence artificielle, telle que décrite dans le titre "Exploration des Paradigmes de l'Intelligence Artificielle : Machine Learning, Deep Learning et Apprentissage par Renforcement", se concentre sur l'étude et la compréhension de différentes approches et techniques utilisées dans le domaine de l'intelligence artificielle.

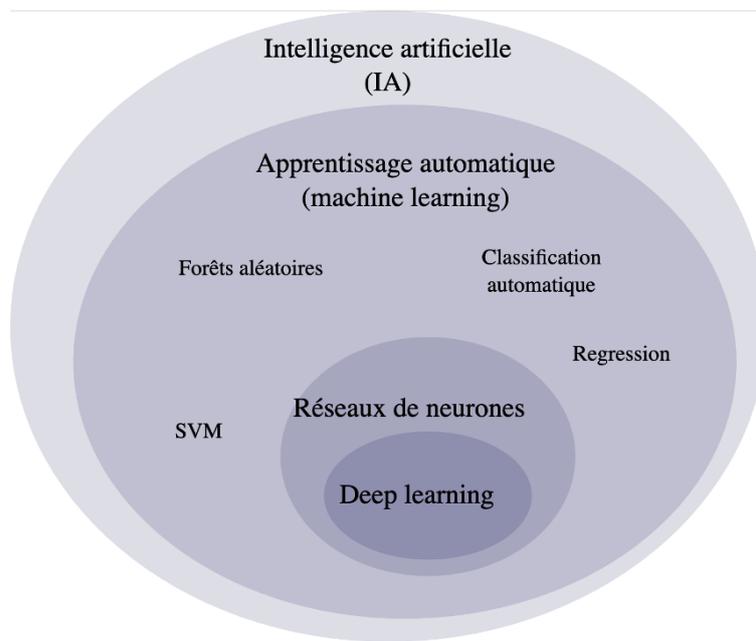


FIGURE 3.1 – Le monde de l'intelligence artificielle

D'où la relation entre le machine learning, le deep learning et l'apprentissage par renforcement est complémentaire et souvent intégrée dans le développement de systèmes intelligents [31].

Le machine learning englobe le deep learning et l'apprentissage par renforcement. Le deep learning utilise des réseaux de neurones pour modéliser des données hiérarchiquement, tandis que l'apprentissage par renforcement consiste à apprendre par interaction avec un environnement. Ces approches peuvent être combinées pour créer des systèmes puissants, par exemple en utilisant le deep learning pour extraire des caractéristiques de données et l'apprentissage par renforcement pour prendre des décisions basées sur ces caractéristiques. Cette intégration permet de relever des défis complexes dans divers domaines comme la robotique et les jeux sociaux.

### 3.3 Deep Learning (Apprentissage profond) :

Le Deep Learning, ou apprentissage profond, est une branche de l'intelligence artificielle qui utilise des réseaux de neurones artificiels pour analyser et apprendre à partir des données [36]. Inspiré du cerveau humain, il excelle dans le traitement de données complexes comme les images et le texte, et est à la base de nombreuses avancées en vision par ordinateur, traitement du langage naturel et reconnaissance vocale.

#### 3.3.1 Introduction aux réseaux de neurones artificiels (ANN)

Les réseaux de neurones artificiels, souvent abrégés en ANN, sont des modèles de calcul inspirés du fonctionnement du cerveau humain. Ils font partie du domaine de l'intelligence artificielle et sont conçus pour apprendre des modèles complexes à partir de données en utilisant des algorithmes d'apprentissage automatique. Un réseau de neurones est composé de couches de neurones interconnectés, et chaque connexion entre les neurones est associée à un poids qui est ajusté pendant le processus d'apprentissage [36].

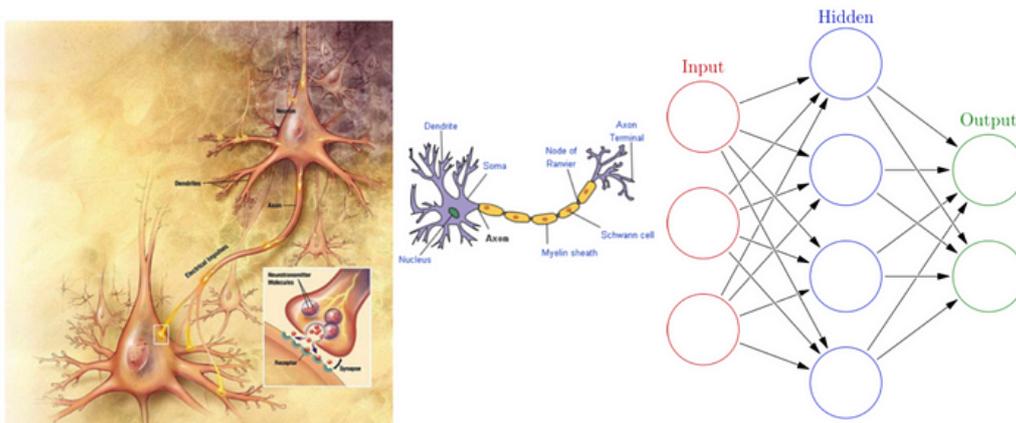


FIGURE 3.2 – Un neurone dans le cerveau humain

#### 3.3.2 Architecture des réseaux de neurones

Les réseaux de neurones artificiels se structurent en trois types de couches : les couches d'entrée, les couches cachées et les couches de sortie [36].

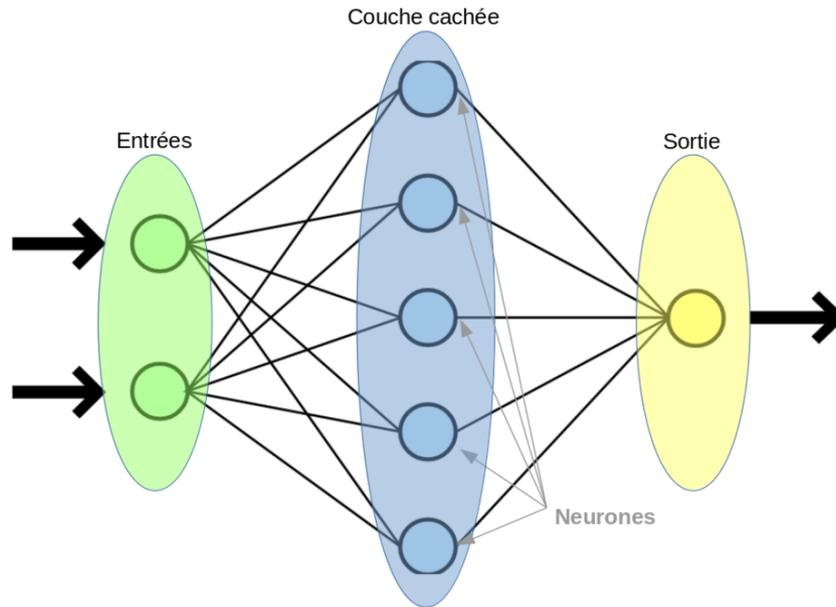


FIGURE 3.3 – Un neurone dans le cerveau humain

1. **Les Couches d'Entrée** : Les couches d'entrée sont chargées de recevoir des entrées et de les transmettre directement à d'autres couches sans effectuer de traitement spécifique.
2. **Les Couches Cachées** : Les couches cachées, bien que facultatives, jouent un rôle crucial dans le traitement des informations. Elles reçoivent les entrées des couches précédentes, effectuent des traitements spécifiques (comme la sommation et la comparaison avec T), puis transmettent les résultats aux couches suivantes pour des traitements ultérieurs.
3. **Les Couches de Sortie** : Les couches de sortie sont responsables de la génération des résultats finaux du réseau. Elles reçoivent les informations traitées par les couches cachées et produisent la sortie finale du réseau.

### 3.3.3 Les différents types de réseaux neuronaux

Les réseaux neuronaux peuvent varier considérablement en termes de nombre de nœuds et de couches, ainsi que dans la manière dont ces nœuds interagissent. Voici une classification générale des principaux types de réseaux neuronaux [36] :

1. **Réseaux neuronaux peu profonds** : Ces réseaux comportent généralement une seule couche cachée. Ils sont rapides et nécessitent moins de puissance de calcul, mais sont limités dans leur capacité à traiter des tâches complexes.
2. **Réseaux neuronaux profonds** : Ces réseaux comportent plusieurs couches cachées. Ils peuvent gérer des tâches beaucoup plus complexes que les réseaux peu profonds, mais exigent plus de ressources et de temps de calcul.

### 3.3.4 Les méthodes d'apprentissage du réseau neuronal artificiel

1. **Apprentissage supervisé** : Dans ce mode, les données sont étiquetées pour indiquer à la machine les patterns à rechercher. Le système s'entraîne sur un ensemble de données étiquetées, apprenant ainsi à associer des entrées spécifiques à des sorties désirées. Cette méthode est couramment utilisée car elle nécessite moins de données d'entraînement et facilite le processus d'entraînement en permettant la comparaison avec des données déjà étiquetées. Cependant, elle peut être coûteuse en raison de l'étiquetage des données et peut conduire à des biais dans le modèle [36].
2. **Apprentissage non-supervisé** : Contrairement à l'apprentissage supervisé, ici les données ne sont pas étiquetées. La machine explore les données à la recherche de patterns sans aucune indication préalable. Elle utilise des algorithmes pour extraire des caractéristiques pertinentes et classer les données en temps réel sans intervention humaine. Bien que moins courante, cette approche est utile pour identifier des patterns et des relations que les humains pourraient ne pas remarquer dans les données, notamment dans le domaine de la cybersécurité [36].
3. **Apprentissage par renforcement** : Cette approche consiste à laisser un algorithme apprendre de ses erreurs pour atteindre un objectif spécifique. L'algorithme explore différentes approches pour atteindre son but et est récompensé ou pénalisé en fonction de ses performances. Cette technique est souvent utilisée pour surpasser les humains dans des jeux complexes, comme le jeu de Go ou Dota 2, où des algorithmes tels qu'AlphaGo de Google et des IA d'OpenAI ont démontré leur supériorité grâce à l'apprentissage par renforcement [36].

### 3.3.5 Les composants fondamentaux d'un neurone artificiel

Dans les réseaux de neurones, le composant fondamental est le neurone artificiel [36], modélisé d'après les observations des neurones biologiques. Le schéma utilisé pour émuler un neurone biologique comprend les éléments suivants :

1. **Poids de Connexion** :  $w_{ij}$  mesure l'influence entre les neurones d'entrée (i) et de sortie (j). Ces poids déterminent l'importance des signaux entrants et sont ajustés pendant l'entraînement du réseau.
2. **Calcul de la Somme Pondérée** :  $net_j = \sum_i w_{ij} \cdot x_i$ . Cette formule représente la somme pondérée des entrées pour le neurone j, où  $x_i$  est l'entrée provenant du neurone i et  $w_{ij}$  est le poids de la connexion entre les neurones i et j.
3. **Fonction d'Activation** :  $O_j = f(net_j)$ . La fonction d'activation transforme la somme pondérée en une sortie, introduisant la non-linéarité nécessaire pour modéliser des relations complexes dans les données. Des exemples de fonctions d'activation incluent ReLU, sigmoïde, et tanh.

4. **État de Sortie** :  $O_j$ , la sortie finale du neurone  $j$  après le calcul de la fonction de transition et l'application de la fonction d'activation. Cette sortie peut ensuite être utilisée comme entrée pour d'autres neurones dans les couches suivantes du réseau.
5. **Formule Globale** :  $O_j = f(\sum_i w_{ij} \cdot x_i)$ . Cette formule est appliquée à chaque neurone pour générer l'activité du réseau en réponse à une entrée. Elle encapsule le processus de réception des entrées pondérées, de leur sommation, et de la transformation via la fonction d'activation.

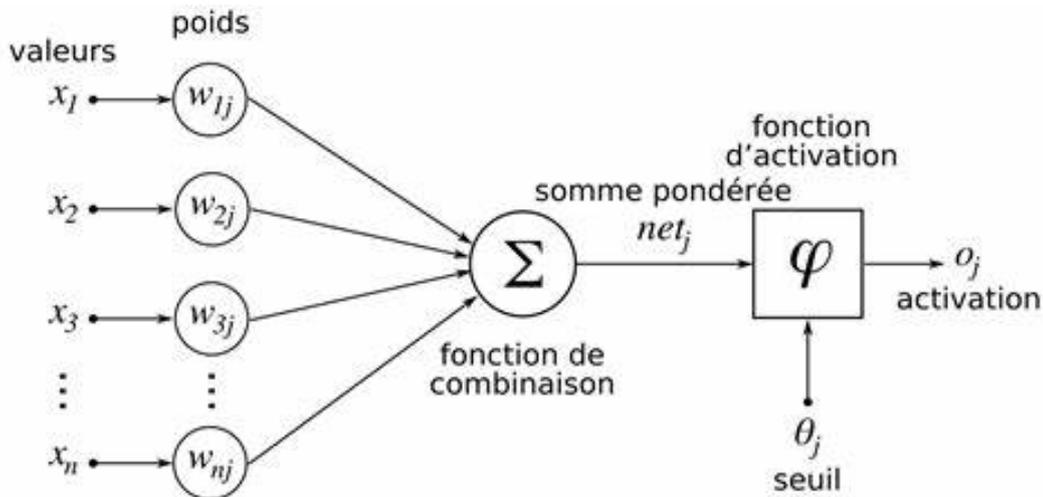


FIGURE 3.4 – Les composant fondamentale d'un neurone artificielle

### 3.3.6 L'entraînement des réseaux de neurones artificiels

1. **Collecte des données** : Rassembler un ensemble de données représentatif de la tâche que le réseau de neurones doit effectuer. Ces données sont généralement divisées en ensembles d'entraînement, de validation et de test.
2. **Prétraitement des données** : Normaliser les données pour les mettre à l'échelle et les rendre comparables. Effectuer des opérations telles que la normalisation des zéros-moyens, et la conversion des données en un format adapté au modèle.
3. **Construction du modèle** : Définir l'architecture du réseau de neurones, y compris le nombre de couches, le nombre de neurones par couche, les fonctions d'activation, etc. Initialiser les poids du réseau.
4. **Définition de la fonction de perte (loss function)** : Choisir une fonction de perte qui mesure la différence entre les prédictions du modèle et les vraies valeurs. L'objectif est de minimiser cette fonction de perte pendant l'entraînement.
5. **Configuration de l'optimiseur** : Sélectionner un algorithme d'optimisation qui ajuste les poids du réseau afin de minimiser la fonction de perte. Des optimiseurs couramment utilisés incluent la descente de gradient stochastique (SGD), Adam, RMSprop, etc.
6. **Entraînement du modèle** : Faire passer les données d'entraînement à travers le réseau de neurones. Calculer la perte (erreur) entre les prédictions du modèle et les vraies valeurs.

Utiliser l'optimiseur pour ajuster les poids du réseau afin de réduire cette perte. Répéter ces étapes sur plusieurs époques (itérations à travers l'ensemble d'entraînement) jusqu'à ce que le modèle atteigne une performance satisfaisante.

7. **Évaluation du modèle** : Utiliser l'ensemble de validation pour évaluer la performance du modèle pendant l'entraînement. Cela permet de surveiller le surajustement (overfitting) et d'ajuster les hyperparamètres si nécessaire.
8. **Test du modèle** : Une fois l'entraînement terminé, utiliser l'ensemble de test pour évaluer les performances du modèle sur des données qu'il n'a jamais vues auparavant.
9. **Fine-tuning (ajustement fin)** : Selon les résultats obtenus sur l'ensemble des tests, il est possible de procéder à des ajustements supplémentaires du modèle, comme l'ajustement des hyperparamètres, la modification de l'architecture, etc

### 3.3.7 Introduction aux Fonctions d'Activation

Les fonctions d'activation sont des fonctions mathématiques non linéaires appliquées à la sortie d'un neurone. Elles permettent de produire une réponse non linéaire à partir des entrées d'un neurone. On utilise généralement différentes fonctions d'activation pour différents types de tâches de modélisation.

Il existe plusieurs fonctions d'activation [36] parmi les plus connues sont :

1. **La fonction ReLU (Rectified Linear Units)** : La fonction ReLU est une fonction de seuil qui produit une sortie linéaire pour les entrées positives et une sortie nulle pour les entrées négatives. Cette fonction permet d'accélérer l'apprentissage dans les réseaux de neurones convolutifs. Mathématiquement :

$$S_j = f(x) = \max(0, x)$$

2. **La fonction d'activation de type « sigmoïde » ou « logistique »** : La fonction sigmoïde est une fonction en S qui est souvent utilisée en raison de sa capacité à produire une réponse relativement douce et à borner les sorties entre 0 et 1. Mathématiquement, cette fonction est définie comme suit :

$$f(x) = \frac{1}{1 + \exp(-x)}$$

3. **La fonction « tangente hyperbolique »** : La fonction tangente hyperbolique est une fonction en S similaire à la sigmoïde, mais elle produit des sorties bornées entre -1 et 1. Cette fonction est souvent utilisée en raison de sa capacité à produire des sorties centrées autour de zéro. Mathématiquement :

$$f(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}$$

4. **La fonction d'activation « softmax »** : La fonction Softmax est une fonction d'activation souvent utilisée pour des tâches de classification. Elle prend en entrée un vecteur et produit

un nouveau vecteur de la même taille en normalisant exponentiellement les valeurs d'entrée. Cela garantit que la somme des valeurs de sortie est égale à 1, ce qui permet d'interpréter les sorties comme des probabilités. Mathématiquement :

$$f(x) = \frac{\exp(x)}{\sum(\exp(x))}$$

Ces fonctions d'activation peuvent être utilisées dans différentes architectures de réseaux de neurones pour produire des sorties non linéaires et donc améliorer les performances du modèle.

### 3.3.8 Les algorithmes dans les réseaux de neurones artificiels

Les algorithmes d'optimisation populaires utilisés dans l'apprentissage profond et les réseaux de neurones sont :

1. **SGD (Stochastic Gradient Descent)** : Une version de la descente de gradient où les gradients sont calculés et appliqués pour chaque échantillon ou mini-lot, ce qui permet une mise à jour plus fréquente des poids et peut mener à une convergence plus rapide et une meilleure généralisation.
2. **Momentum** : Ajoute une fraction de la mise à jour précédente à la mise à jour actuelle, accélérant ainsi les trajectoires de descente dans les dimensions avec gradients persistants tout en atténuant les oscillations.
3. **Nesterov Accelerated Gradient (NAG)** : Similaire à Momentum, mais calcule d'abord le gradient par rapport à une estimation future des paramètres, fournissant ainsi une correction supplémentaire pour des mises à jour plus précises.
4. **Adagrad** : Adapte le taux d'apprentissage pour chaque paramètre en fonction de la fréquence et de l'importance des mises à jour passées, augmentant ainsi le taux d'apprentissage pour les paramètres qui changent rarement.
5. **Adadelta** : Amélioration d'Adagrad qui limite la somme des gradients accumulés à une fenêtre fixe, atténuant ainsi le problème de diminution agressive des taux d'apprentissage en Adagrad.
6. **RMSprop** : Maintient une moyenne mobile exponentielle des carrés des gradients et adapte les taux d'apprentissage en conséquence, équilibrant ainsi les taux d'apprentissage pour des mises à jour stables.
7. **Adam (Adaptive Moment Estimation)** : Combine les avantages de RMSprop et de Momentum en maintenant une moyenne mobile des premiers et des seconds moments des gradients, ajustant également les taux d'apprentissage de manière adaptative.
8. **AdaMax** : Variante d'Adam utilisant la norme infinie plutôt que la norme L2 pour la mise à jour des moments.

9. **Nadam** : Version d'Adam qui incorpore les avantages de NAG pour améliorer les mises à jour des paramètres.

### 3.3.9 Intégration du Biais

Le biais [36] peut être représenté par un paramètre supplémentaire  $b$  ajouté à chaque neurone. Pour calculer la sortie d'un neurone, on ajoute le biais au produit scalaire des entrées et des poids, avant l'application de la fonction d'activation. Mathématiquement, cela peut être représenté comme suit :

$$z = b + \sum(w \cdot x)$$
$$O = f(z)$$

où :

- $b$  est le biais ajouté au neurone,
- $x$  est le vecteur d'entrée,
- $w$  est le vecteur de poids associé à chaque entrée,
- $z$  est la somme pondérée des entrées et des poids avec le biais ajouté,
- $f$  est la fonction d'activation appliquée à la sortie des neurones,
- $O$  est la sortie du neurone.

### 3.3.10 Gestion de l'Erreur

L'erreur mesure la différence entre la sortie réelle d'un modèle et celle attendue pour un ensemble donné d'entrées, crucial pour évaluer les performances du modèle et ajuster ses paramètres [36]. Elle peut être mesurée de diverses manières selon la tâche. L'analyse de l'erreur aide à diagnostiquer des problèmes comme le sous-ajustement ou le sur-ajustement. Minimiser l'erreur est l'objectif, souvent réalisé par des techniques telles que la rétropropagation du gradient ou la descente de gradient stochastique. Le biais, représenté par un paramètre supplémentaire  $b$ , est ajouté à chaque neurone, influençant la sortie via le produit scalaire des entrées et des poids, avant l'activation. Mathématiquement, cela s'exprime ainsi :

$$z = b + \sum(w \cdot x)$$
$$y = f(z)$$

où :

- $b$  est le biais ajouté au neurone,
- $x$  est le vecteur d'entrée,
- $w$  est le vecteur de poids associé à chaque entrée,
- $z$  est la somme pondérée des entrées et des poids avec le biais ajouté,
- $f$  est la fonction d'activation appliquée à la sortie des neurones,

### 3.4 Reinforcement Learning (Apprentissage par renforcement)

L'apprentissage par renforcement [41] est une branche de l'intelligence artificielle qui s'inspire du comportement des organismes vivants pour apprendre à prendre des décisions dans des environnements dynamiques. Dans ce paradigme, un agent interagit avec un environnement en prenant des actions et en recevant des récompenses ou des punitions en fonction de ses actions. L'objectif de l'agent est de maximiser les récompenses cumulées au fil du temps en apprenant une politique, c'est-à-dire une stratégie de prise de décision, qui lui permet d'atteindre ses objectifs.

#### 3.4.1 Composants Principaux

Dans le domaine de l'apprentissage par renforcement, plusieurs concepts fondamentaux définissent les interactions entre un agent et son environnement. Comprendre ces concepts est essentiel pour saisir le fonctionnement de cette approche d'intelligence artificielle. Explorons donc brièvement ces notions clés avant de plonger plus profondément dans l'apprentissage par renforcement. Dans le domaine de l'apprentissage par renforcement, plusieurs concepts fondamentaux [41] définissent les interactions entre un agent et son environnement. Comprendre ces concepts est essentiel pour saisir le fonctionnement de cette approche d'intelligence artificielle. Explorons donc brièvement ces notions clés avant de plonger plus profondément dans l'apprentissage par renforcement.

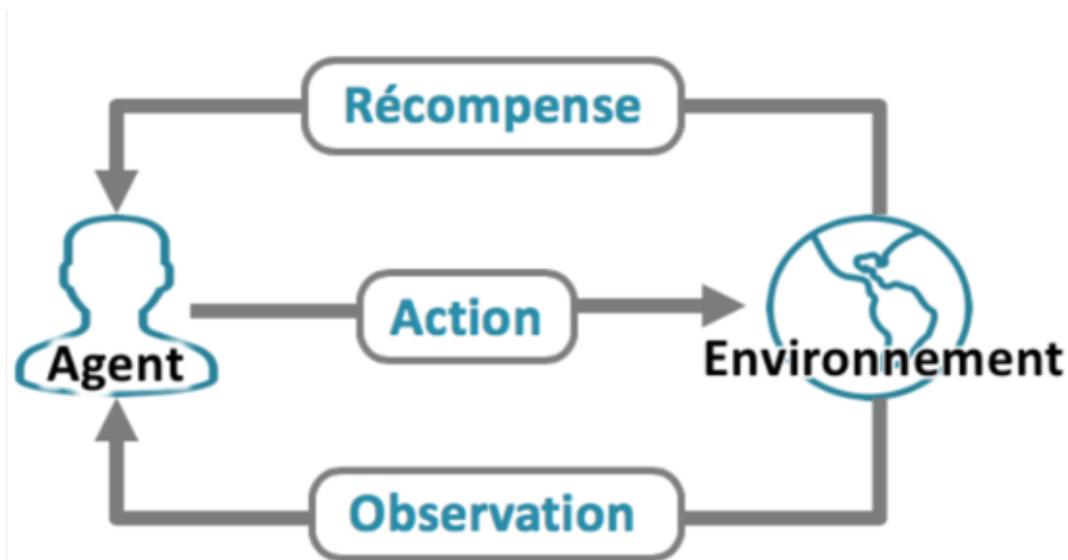


FIGURE 3.5 – Composants *Q-learning*

1. **Agent** : Le joueur ou le preneur de décision qui perçoit l'environnement extérieur et agit en conséquence, il détecte la présence de véhicules, et ajuste les feux de signalisation en conséquence pour optimiser la fluidité du trafic et réduire les temps d'attente.
2. **Environnement** : Le monde extérieur avec lequel interagit l'agent, qui agit en retour vis-à-vis des actions prises par l'agent, comme les véhicules, les intersections et le réseaux routier .

3. **État** : L'information utile à l'agent, observée et notée  $s$ . Les actions possibles peuvent être choisies en fonction des états, comme le nombre de véhicules en attente à chaque direction d'une intersection, vitesse moyenne des véhicules, temps écoulé depuis le dernier changement de feu.
4. **Action** : Le comportement qui affecte l'environnement, noté  $a$ . Dans ce cas, l'action serait de changer l'état des feux de signalisation, par exemple, passer le feu au vert pour permettre aux véhicules d'avancer dans une direction particulière.
5. **Récompense** : Le gain numérique acquis par l'agent après réalisation de l'action sur l'état courant, et noté  $r$ . Une récompense positive pourrait être attribuée pour chaque seconde de réduction du temps d'attente moyen des véhicules, ou pour l'augmentation du flux de trafic sans engendrer de congestion.
6. **Transition** : La fonction de transition ( $Tr$ ) qui définit comment l'agent passe d'un état à un autre en effectuant une action donnée. Par exemple, si le feu devient vert, les véhicules dans cette voie commencent à avancer, modifiant ainsi l'état initial (nombre de véhicules en attente) vers un nouvel état.

### 3.4.2 L'Algorithmes d'apprentissage par renforcement

Les algorithmes clés comme le Q-learning, les Deep Q-Networks (DQN) et les méthodes de Policy Gradient sont essentiels dans ce domaine, ayant démontré leur efficacité dans divers domaines d'application. Explorons maintenant l'algorithme Q-learning pour mieux comprendre son fonctionnement et ses utilisations. [41].

## L'algorithme Q-learning adapté aux environnements discrets de petite taille

Le Q-learning est un algorithme d'apprentissage par renforcement qui permet à un agent d'apprendre à prendre des décisions séquentielles dans un environnement dynamique afin de maximiser les récompenses cumulées au fil du temps.

### 1. Mécanisme du Q-learning :

La fonction de transition ( $Tr$ ) permet à l'agent de passer d'un état à un autre en suivant une politique et en choisissant une action  $a \in A$ . Quand l'agent accomplit une action dans un état donné, il est récompensé. La principale mission de l'agent consiste à acquérir la capacité de maximiser les récompenses en utilisant les valeurs stockées dans la table  $Q$ .

### 2. Calcul de la récompense à long terme :

La détermination de la récompense à long terme, généralement connue sous le nom de  $R_t$ , est une étape essentielle dans le processus d'apprentissage par renforcement afin d'évaluer les conséquences à long terme des actions de l'agent. Cette récompense à long terme est

généralement calculée en prenant en compte les récompenses obtenues à chaque étape  $t$ , pondérées par un facteur de discount  $\gamma$  qui représente l'importance relative des récompenses futures par rapport aux récompenses immédiates.

Voici la façon de calculer la récompense à long terme :

- (a) **Récompense instantanée** : À chaque étape  $t$ , l'agent reçoit une récompense instantanée  $r_t$ .
- (b) **Le facteur de réduction  $\gamma$**  : Le facteur  $\gamma (0 \leq \gamma \leq 1)$  dit facteur de discount est utilisé pour pondérer les récompenses futures par rapport aux récompenses immédiates. Il représente l'importance relative des récompenses futures par rapport à la récompense actuelle. Un facteur de réduction élevé accorde plus d'importance aux récompenses futures, tandis qu'un facteur de discount faible les néglige davantage.  
Un  $\gamma$  élevé accorde plus d'importance aux récompenses futures, favorisant une exploration à long terme, tandis qu'un  $\gamma$  faible privilégie les récompenses immédiates et une stratégie plus orientée vers l'exploitation.
- (c) **Récompense à long terme** : La récompense à long terme  $R_t$  est déterminée en ajoutant les récompenses futures pondérées par le facteur de réduction  $\gamma$  jusqu'à la fin de l'horizon temporel ou jusqu'à la convergence de l'algorithme, selon le cas [7] :

$$R_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots + \gamma^{n-t} r_{n-1} \quad (3.1)$$

où :

- $r_t$  : est la récompense obtenue à l'étape  $t$ .
- $\gamma$  : est le facteur de discount.
- $n$  : est le nombre total d'étapes.  $t$  est l'étape actuelle.

3. **Mise à jour de la valeur  $Q$**  : La valeur  $Q$  qui représente la paire  $(s, a)$  représente la qualité de l'action  $a$  dans l'état donné  $s$ . Elle est évaluée en utilisant la fonction  $Q$  de la manière suivante [7] :

$$Q_{\text{new}}(s_t, a_t) = (1 - \alpha) \times Q(s_t, a_t) + \alpha \times [r + \gamma \times \max_a Q(s_{t+1}, a)] \quad (3.2)$$

Où :

- $s_t$  : est l'état actuel.
- $s_{t+1}$  : est l'état suivant (futur).

- $Q(s_t, a_t)$  : et  $Q_{\text{new}}(s_t, a_t)$  sont les anciennes et les nouvelles valeurs de la valeur  $Q$  correspondant à l'état  $s_t$  et à l'action  $a_t$ , respectivement.

- $\alpha$  : est le taux d'apprentissage et  $\gamma$  est le facteur discount.

Les étapes de l'algorithme  $Q$ -learning sont décrites dans l'algorithme 2,

---

**Algorithm 2**  $Q$ -learning

---

**Input:** Processus d'états aléatoires

**Output:** Table  $Q$

**Début**

Initialiser aléatoirement les valeurs  $Q(s, a)$

**repeat**

Initialiser l'état  $s$

Sélectionner l'action  $a$  à partir de  $s$  en utilisant la politique dérivée de  $Q$  (par exemple,  $\varepsilon$ -greedy avec  $\varepsilon \in [0, 1]$ )

Exécuter l'action  $a$ , et observer la récompense  $r$  ainsi que le nouvel état  $s_{t+1}$

Mettre à jour  $Q(s, a)$  en utilisant l'équation 3.2;

$s \leftarrow s_{t+1}$

**until** L'état  $s$  soit terminal;

Retourner Table  $Q$

---

### 3.5 Problématique

La régulation des feux de signalisation urbains basée sur la théorie des jeux non coopératifs présente des avantages potentiels pour optimiser le flux de trafic à travers les intersections. Cependant, cette approche rencontre des défis significatifs, notamment la dépendance critique à l'égard des données précises de trafic collectées par des capteurs tels que les boucles d'induction. En l'absence ou en cas de dysfonctionnement de ces collecteurs de données, le système peut perdre en efficacité, compromettant ainsi sa capacité à réagir de manière adaptative aux conditions en temps réel.

Comment améliorer l'efficacité de ce système et quelle solution envisager en l'absence de ces collecteurs de données ?

Dans cette partie, nous présentons un système de contrôle adaptatif des feux de signalisation basé sur le modèle MetMod et l'apprentissage par renforcement, que nous avons développé dans le cadre de ce mémoire .

### 3.6 L'apprentissage par renforcement(RL) et la théorie des jeux (TJ)

RL et la TJ sont deux domaines clés de l'intelligence artificielle. L'apprentissage par renforcement consiste à apprendre à prendre des décisions dans des environnements dynamiques en interagissant avec ceux-ci et en recevant des récompenses ou des punitions. La TJ, quant à elle, étudie les interactions stratégiques entre des agents rationnels et les résultats de ces interactions. Ces deux domaines sont souvent combinés pour résoudre des problèmes complexes de prise de décision dans des environnements dynamiques et incertains.

### 3.7 Principe général du jeu avec RL proposé

Dans cette section, nous présentons notre approche basée sur le RL pour le contrôle adaptatif des feux de circulation, désignée ci-après MetMod-RL . Ce modèle se compose de deux couches distinctes :

- La première couche utilise la théorie des jeux pour modéliser les interactions entre les agents et les stratégies possibles pour chacun d'eux. Chaque agent vise à optimiser ses résultats en fonction de ses propres objectifs, tout en tenant compte des actions des autres agents.
- La deuxième couche exploite l'algorithme d'apprentissage par renforcement Q-learning pour améliorer la prise de décision des agents. Dans ce contexte, les joueurs sont considérés comme des agents d'apprentissage autonomes, chacun cherchant à minimiser son propre objectif en choisissant la meilleure stratégie parmi un ensemble prédéfini de stratégies. Les agents explorent de manière itérative les différentes actions et les récompenses associées, leur permettant d'adapter leurs choix en fonction des résultats passés et d'optimiser progressivement leurs performances pour le contrôle adaptatif des feux de circulation.

Le modèle MetMod-RL se définit de la manière suivante :

$$\mathbf{MetMod-RL} = \{P_1, P_2, P_3, P_4, A_{P_1}, A_{P_2}, A_{P_3}, A_{P_4}, R_{P_1}, R_{P_2}, R_{P_3}, R_{P_4}, \alpha, \gamma\}$$

où :

- $P_1, P_2, P_3, P_4$  :représentent les quatres agents d'apprentissage, visant respectivement à minimiser le nombre de véhicules sur les liens de sa phase.
- $A_{P_1}, A_{P_2}, A_{P_3}, A_{P_4}$  : représentent les actions de chaque agents, qui consiste à choisir la durée de feu vert .
- $R_{P_1}, R_{P_2}, R_{P_3}, R_{P_4}$  : représentent les fonctions de récompences de chaque agent. Elle reflète l'impact direct (coût) sur la solution lorsqu'un agent  $i \in I$  effectue une action a de son ensemble  $A_{P_i}$  d'actions .
- $\alpha, \gamma$  : sont les taux d'apprentissage et de discount, respectivement.

**Le principe général de MetMod-RL est décrit par l'algorithme 3**

Le jeu se déroule en **max-ep** épisodes. Chaque épisode représente un cycle de feu de circulation. À chaque étape, les agents doivent déterminer le nombre de véhicules ( $N_i$ ) présents dans la phase au début du cycle, le débit de sortie ( $S_i$ ) des véhicules de la phase pendant le cycle et le débit d'arrivée ( $E_i$ ) des véhicules dans la phase pendant le cycle, en exécutant une action ( $g_i$ ) parmi un ensemble prédéfini d'actions, chaque agent se réfère à sa base de connaissances, représentée par une Q-table. Lors des premières étapes d'apprentissage, les éléments de la Q-table sont initialisés à zéro, ce qui implique qu'aucune information préalable concernant l'environnement n'est disponible.

---

**Algorithm 3** MetMod-RL

---

**Input:**  $E_i, S_i, N_i, g_i, max - ep, Q - table$

**Output:** l'ensemble **Sols** des **max-ep** solutions construites

**Début**

1. Initialiser les éléments de **Q-table** à **0**
2. Pour **i** allant de **1** à **max-ep** Faire :
3. **Sol(i)** =  $\emptyset$
4. Choisir aléatoirement un agent
5. Choisir aléatoirement un état  $s_t$  de l'ensemble  $S$
6. Choisir aléatoirement une action  $a_t$  de l'ensemble  $A$
7. Calculer la récompense  $r$  correspondant à la combinaison (état ; action) :  $Q(s_t, a_t)$
8. Déterminer le prochain état  $s_{t+1}$
9. Déterminer le prochain action  $s_{t+1}$
10. Mettre à jour  $Q(s_t, a_t)$
11. Mettre à jour la solution actuelle **Sol(i)**
12. FIN Pour

Retourner **Sols**

---

**1. Etape 01 : L'état et L'action**

- **L'état** : L'état forme une représentation de l'environnement à un instant  $t$  noté  $s_t$ . Il est modélisé dans notre modèle par  $E_i, S_i$  et  $N_i$ .
- **L'action** : une action de l'agent représente la durée adaptative de feu vert de chaque phase à chaque instat  $t$  noté  $a_t$ .

**2. Etape 02 : La récompense**

L'agent d'apprentissage  $i$  utilise sa fonction de récompense, notée  $R_i$ , pour évaluer la valeur de la récompense, notée  $r_i$ . Cette fonction permet d'évaluer la qualité de la solution en termes de coût (interférence ou séparation) par rapport à la qualité précédente avant l'exécution de l'action  $a$ .

- La fonction de récompense  $R_i^{(1)}$  est définie comme suit :

$$R_i^{(1)} = \begin{cases} +1, & \text{si les trois contraintes (2.8) (2.9) et (2.10) sont satisfaites} \\ -1, & \text{sinon} \end{cases} \quad (3.3)$$

- La fonction de récompense  $R_i^{(2)}$  est définie comme suit :

$$R_i^{(2)} = N_i + E_i \sum_{j=1, j \neq i}^{PJ} g_j + (E_i - S_i)g_i \quad (3.4)$$

Chaque agent cherche à minimiser le nombre de véhicules restant dans ça phase a la fin de chaque cycle, qui est la fonction objectif dans le modèle MetMod (2.13)

### 3. Etape 03 : Détermination du prochain état $s_{t+1}$

- **Cas de la présence des collecteurs de données** : Pour obtenir le prochain état  $s_{t+1}$ , les boucles d'induction collectent les taux d'arrivée et de départ des véhicules, ainsi que le nombre de véhicules présents dans chaque phase à la fin de chaque cycle. Ensuite, une action est simulée, comme l'ajustement de la durée des feux verts, et les données sont à nouveau collectées. Le prochain état  $s_{t+1}$  est déterminé par les nouveaux taux d'arrivée et de départ, ainsi que le nouveau nombre de véhicules présents après la simulation de l'action. Ces informations permettent de voir l'impact de l'action sur le trafic et d'ajuster les décisions futures.

- **Cas de l'absence des collecteurs de données** : Lorsque les boucles d'induction ne sont pas disponibles, on peut utiliser des prédictions avec des réseaux de neurones profonds pour déterminer le prochain état  $s_{t+1}$ . Pour cela on a :

- En absence de données directes des boucles d'induction, utilisez d'autres sources de données disponibles, comme les données historiques de trafic.
- Préparez les données en les nettoyant, en les transformant et en les normalisant pour les rendre appropriées à l'entrée du modèle de réseau de neurones.
- Concevez et entraînez un réseau de neurones profond adapté à votre problème spécifique. Cela peut inclure plusieurs couches cachées avec des fonctions d'activation .
- Utilisez les données prétraitées pour entraîner le modèle. Les réseaux de neurones profonds apprennent à partir des données historiques pour prédire les comportements futurs du trafic.
- Une fois le modèle entraîné, utilisez-le pour prédire le prochain état  $s_{t+1}$  en fournissant les données actuelles  $s_t$ . Le modèle génère des prédictions basées sur les modèles appris à partir

des données d'entraînement.

#### 4. Etape 04 : Détermination de la prochain action $a_{t+1}$

Pour sélectionner l'action  $a_{t+1}$  de l'état  $s_{t+1}$  en utilisant une politique dérivée de  $Q$  (par exemple,  $\epsilon$ -greedy), nous pouvons procéder de la manière suivante en tenant compte de la présence ou de l'absence des collecteurs de données :

- **Cas de la présence des collecteurs de données :**

Lorsque les collecteurs de données sont disponibles, le processus commence par la collecte d'informations précises sur l'état actuel  $s_t$  à l'aide des boucles d'induction .

Ensuite, les valeurs  $Q$  sont calculées à partir de la  $Q$ -table, permettant ainsi de déterminer les meilleures actions possibles à entreprendre à partir de  $s_t$ .

Pour décider de l'action à exécuter, une politique  $\epsilon$ -greedy est utilisée : avec une probabilité  $\epsilon$ , une action est choisie au hasard pour explorer de nouvelles possibilités, tandis qu'avec une probabilité  $1 - \epsilon$ , l'action maximisant la valeur  $Q$  est sélectionnée pour exploiter les connaissances acquises.

Après l'exécution de l'action choisie, les collecteurs de données sont à nouveau sollicités pour observer l'état résultant  $s_{t+1}$ , qui devient alors le nouvel état de référence pour le prochain cycle de prise de décision.

- **Cas où il y a de l'absence des collecteurs de données :**

En absence de collecteurs de données fonctionnels, des sources alternatives comme des données historiques, sont utilisées pour estimer l'état actuel  $s_t$ .

Un modèle de prédiction tel qu'un réseau de neurones est souvent employé pour estimer  $s_{t+1}$  en fonction des données disponibles. Comme dans le cas précédent, les valeurs  $Q$  sont calculées en utilisant l'état  $s_t$  ou l'estimation  $\hat{s}_t$ , et une politique  $\epsilon$ -greedy est appliquée pour choisir l'action suivante : exploration avec probabilité  $\epsilon$  ou exploitation avec probabilité  $1 - \epsilon$ . Après l'exécution de l'action, le modèle de prédiction est utilisé pour estimer le nouvel état  $s_{t+1}$ , qui est ensuite utilisé comme référence pour sélectionner l'action à prendre au prochain cycle.

- **Politique du choix d'action  $\epsilon$ -greedy :**

La politique  $\epsilon$ -greedy est une stratégie utilisée en apprentissage par renforcement pour équilibrer exploration et exploitation lors de la prise de décision par un agent. Cette méthode permet à l'agent de choisir entre deux options [50] :

- **Exploration  $\epsilon$  :** C'est la proportion de temps où l'agent choisit une action de manière aléatoire, sans tenir compte de la valeur estimée de l'action. Cela permet à l'agent de découvrir de nouvelles possibilités et de s'aventurer dans des actions qui pourraient potentiellement conduire à de meilleures stratégies à long terme. Typiquement,  $\epsilon$  est un petit pourcentage,

par exemple 10%.

- **Exploitation  $1-\epsilon$**  : C'est la proportion de temps où l'agent choisit l'action avec la meilleure valeur estimée jusqu'à présent. Cela permet à l'agent de maximiser la récompense immédiate en se basant sur les connaissances acquises à travers l'expérience. Ainsi,  $1-\epsilon$  correspond à une probabilité plus élevée, par exemple 90%.

La politique  $\epsilon$ -greedy en apprentissage par renforcement utilise des probabilités pour que l'agent explore de nouvelles actions avec un petit pourcentage  $\epsilon$ , et exploite les actions connues avec une probabilité de  $1-\epsilon$ . Cela assure un équilibre entre exploration pour découvrir de nouvelles stratégies et exploitation pour maximiser les gains immédiats, adaptant ainsi les décisions de l'agent en fonction des informations disponibles et du paramètre spécifié.

Ces pourcentages sont choisis de manière à garantir que l'agent explore suffisamment pour découvrir de nouvelles options potentiellement avantageuses (exploration), tout en exploitant les actions déjà connues pour maximiser les gains à court terme (exploitation). Cette stratégie aide à éviter de rester coincé dans des stratégies sous-optimales en continuant à explorer de nouvelles alternatives tout en exploitant les actions les plus prometteuses.

#### 5. Etape 05 :

Mise à jour de la valeur actuelle de Q-learning, notée  $Q(s_t, a_t)$ , en utilisant l'équation de Bellman (3.2)

$$Q_{\text{new}}(s_t, a_t) = (1 - \alpha) \times Q(s_t, a_t) + \alpha \times [r + \gamma \times \max_a Q(s_{t+1}, a)]$$

### 3.8 Le Processus d'Apprentissage

- Le processus d'apprentissage du jeu dans le modèle MetMod-RL se déroule sur une série d'épisodes, chacun représentant un cycle de feux de circulation.

- Le processus d'apprentissage se termine après un nombre prédéfini d'épisodes, noté **max-ep**, où chaque épisode constitue un cycle complet de décision et d'action pour les agents. Pendant ces **max-ep** épisodes, un ensemble de **max-ep** solutions est généré, chacune correspondant aux décisions prises par les agents à chaque étape du cycle de feux.

- Les solutions construites au cours de ces épisodes sont évaluées selon des critères prédéfinis, tels que la réduction du nombre de véhicules en attente, le flux de trafic, ou d'autres indicateurs de performance. À la fin de tous les épisodes, les solutions sont comparées en termes de performance. Ces performances se réfèrent à plusieurs critères évalués sur les so-

lutions générées pendant les épisodes :

- Réduction du nombre de véhicules en attente.
- Flux de trafic amélioré.
- Autres mesures de performance définies préalablement.

- L'algorithme a la capacité de choisir la solution la plus optimale en se basant sur l'équilibre de Nash. Un équilibre de Nash se produit lorsque l'agent ne peut améliorer son résultat qu'en modifiant sa propre stratégie, tant que les autres agents ne modifient pas les leurs. On examine les différentes solutions afin de repérer celles qui correspondent à des équilibres de Nash, en comparant les résultats des différentes stratégies et en vérifiant la stabilité des décisions des agents.

- L'algorithme a également la capacité de générer un ensemble de solutions de compromis, appelé front de Pareto, qui regroupe les solutions non dominées qui optimisent simultanément plusieurs critères. Les solutions sont répertoriées en fonction de leur position dominante. On considère qu'une solution est dominante lorsqu'elle est supérieure ou égale dans tous les critères à une autre solution, et strictement supérieure dans au moins un critère supplémentaire.

- À la fin de tous les épisodes, l'algorithme renvoie la meilleure solution trouvée selon les critères de sélection, offrant une approche optimale ou un ensemble de solutions équilibrées pour le contrôle adaptatif des feux de circulation. La meilleure solution est déterminée soit par l'équilibre de Nash pour une performance optimale dans un contexte de jeu stratégique, soit par le front de Pareto pour des solutions de compromis efficaces.

### 3.9 Séparation des Q-Tables par Agent

Dans cette approche spécifique, chaque agent possède sa propre table d'apprentissage. Cela se traduit par l'existence de quatre  $Q$  - tables distinctes pour chaque intersection :

- $Q$  - table -  $P_1$  pour l'agent  $P_1$  .
- $Q$  - table -  $P_2$  pour l'agent  $P_2$  .
- $Q$  - table -  $P_3$  pour l'agent  $P_3$  .
- $Q$  - table -  $P_4$  pour l'agent  $P_4$  .

Aucun des agents n'a la possibilité de modifier la table d'apprentissage de l'autre dans ce cadre.

Cette séparation des Q-tables offre plusieurs avantages :

- Chaque agent a la possibilité de se spécialiser dans l'amélioration des feux de circulation

pour sa propre phase, ce qui peut conduire à des stratégies plus performantes et adaptées à des conditions particulières.

- La réduction de la complexité computationnelle se fait en restreignant les interactions entre les tables d'apprentissage, ce qui permet de rendre l'algorithme plus rapide et plus facile à gérer.
- En cas de panne ou de performance médiocre d'un agent, les autres agents continuent de fonctionner de manière normale, garantissant ainsi une certaine stabilité au système global.

## **Conclusion**

Ce chapitre explore comment l'intelligence artificielle, incluant l'apprentissage supervisé, non supervisé, et par renforcement, ainsi que l'utilisation des réseaux de neurones, transforme la gestion des feux de signalisation urbains. Ces avancées visent à améliorer la fluidité du trafic et à réduire les embouteillages en permettant une adaptation dynamique des systèmes de signalisation aux conditions en temps réel. Le modèle MetMod-RL illustre cette approche en intégrant l'apprentissage par renforcement et la théorie des jeux pour optimiser le flux de trafic urbain, même sans données en temps réel, en utilisant des prédictions basées sur des réseaux de neurones.

## Chapitre 4

# Simulation et Implémentation

### Introduction

Le chapitre qui suit se concentre sur la simulation et l'implémentation de stratégies avancées pour le contrôle des feux de signalisation dans les environnements urbains. En explorant différentes approches telles que MetMod à 2-phases, MetMod à 4-phases, Cournot, et l'introduction de MetMod-RL utilisant des réseaux de neurones, cette étude vise à évaluer et comparer leurs performances respectives. Ces analyses sont essentielles pour comprendre comment ces technologies peuvent optimiser la gestion du trafic, améliorer la fluidité des déplacements, et répondre aux défis croissants de la mobilité urbaine .

### 4.1 Présentation du logiciel SUMO



FIGURE 4.1 – Logo logiciel SUMO

SUMO (Simulation of Urban MObility) est un logiciel open-source spécialisé dans la simulation du trafic routier urbain. Il offre une plateforme flexible pour modéliser et évaluer divers scénarios de transport urbain. Avec SUMO, les utilisateurs peuvent créer des environnements de simulation détaillés, comprenant des réseaux routiers complexes, des intersections, des feux de signalisation et des itinéraires pour les véhicules et les piétons.

L'un des principaux avantages de SUMO est sa capacité à générer des données réalistes sur le trafic, permettant aux chercheurs, planificateurs urbains et ingénieurs de tester différentes stratégies de gestion du trafic, d'évaluer les performances des infrastructures routières et d'analyser l'impact de diverses interventions sur la circulation.

Avec son interface conviviale, SUMO facilite la création, l'exécution et l'analyse des simulations. Les résultats produits par SUMO peuvent fournir des informations précieuses sur les temps de trajet, les niveaux de congestion, les émissions de CO<sub>2</sub> et d'autres métriques liées à la circulation routière, aidant ainsi à prendre des décisions éclairées pour améliorer la mobilité urbaine.

## 4.2 Composants des Scénarios dans SUMO

SUMO offre aux utilisateurs la possibilité d'incorporer une vaste gamme de variations du trafic routier, permettant ainsi d'aborder un large éventail de sujets de gestion du trafic. Voici quelques-unes des fonctionnalités clés :

- SUMO gère divers modes de transport terrestre, facilitant l'intégration et l'analyse des interactions entre eux.
- Le logiciel utilise des modèles avancés pour simuler le mouvement des piétons et leurs interactions avec les véhicules, offrant une représentation réaliste des dynamiques urbaines.
- SUMO permet de définir les caractéristiques des véhicules avec précision, y compris la vitesse, l'accélération, la taille et la forme.
- Les trajets des véhicules sont décrits de manière détaillée, incluant les horaires et lieux de départ et d'arrivée, les itinéraires, les temps d'attente, les variations de vitesse et les changements de voie.
- Les parcours peuvent être spécifiés au niveau des flux de véhicules ou pour chaque véhicule individuellement, offrant une flexibilité dans la simulation et l'analyse.
- SUMO permet de créer des infrastructures routières par insertion personnalisée des composants ou par importation de cartes routières satellitaires aux formats .osm ou autres.
- Le logiciel intègre et respecte les règles de circulation et de priorité selon les codes de la route, au niveau des routes, des intersections, des feux de signalisation, etc., sauf si le scénario exige des exceptions.

## 4.3 Construction d'un Réseau Routier sur SUMO avec un Fichier XML

### 1. Définition des nœuds (intersections) et des arêtes (routes) :

- **Identification des coordonnées spatiales des nœuds** : Déterminez les positions géographiques des intersections sur votre réseau routier. Chaque nœud devrait avoir des coordonnées spécifiques pour sa localisation.

- **Attribution d'un ID unique** : Chaque nœud doit être identifié par un ID unique qui le distinguera des autres nœuds du réseau.

- **Attribution de types aux nœuds** : Si nécessaire, spécifiez des types pour les nœuds, comme "priorité" ou "feu de signalisation", en fonction de leurs caractéristiques sur la route.

- **Création des arêtes entre les nœuds** : Reliez les nœuds en spécifiant les arêtes (routes) qui les relient. Pour chaque arête, identifiez le nœud de départ et le nœud de fin.

- **Inclusion d'attributs supplémentaires** : Ajoutez d'autres attributs aux arêtes comme le type de route, le nombre de voies, etc., pour définir plus précisément les caractéristiques du réseau.

### 2. Génération des Fichiers XML : On doit créer trois fichiers XML distincts :

- **Fichier de nœuds (.nod.xml)** : Ce fichier décrit les nœuds du réseau, y compris leurs IDs, leurs coordonnées et leurs types.

- **Fichier d'arêtes (.edg.xml)** : Ce fichier spécifie les connexions entre les nœuds en définissant les arêtes du réseau.

- **Fichier de types d'arêtes (.type.xml)** : Ce fichier définit les caractéristiques des différentes types de routes, telles que le nombre de voies, la vitesse maximale autorisée, etc.

```
<nodes>
  <node id="n1" x="-500" y="0" type="priority"/>
  <node id="n2" x="-250" y="0" type="traffic_light"/>
  <node id="n3" x="-150" y="200" type="traffic_light"/>
  <node id="n4" x="0" y="0"/>
  <node id="n5" x="150" y="200"/>
</nodes>
```

FIGURE 4.2 – Configuration des nœuds

```

<edges>
  <edge from="n1" to="n2" id="in" type="3L45"/>
  <edge from="n2" to="n3" id="2to3" type="2L15"/>
  <edge from="n3" to="n4" id="3to4" type="3L30"/>
  <edge from="n4" to="n5" id="out" type="3L30"/>
</edges>

```

FIGURE 4.3 – Configuration des arêtes

```

<types>
  <type id="3L45" priority="3" numLanes="3" speed="45"/>
  <type id="2L15" priority="3" numLanes="2" speed="15"/>
  <type id="3L30" priority="2" numLanes="3" speed="30"/>
</types>

```

FIGURE 4.4 – Configuration des types

Il faut assurer que chaque fichier XML contient toutes les informations nécessaires, telles que les priorités, les limites de vitesse et le nombre de voies pour chaque type de route. Ces attributs permettent de définir les caractéristiques spécifiques du réseau routier.

### 3. Génération du Réseau avec netconvert :

- **Combinaison des fichiers XML avec netconvert** : Utilisez l'outil netconvert pour fusionner les fichiers de nœuds, d'arêtes et de types d'arêtes en un seul fichier de réseau (.net.xml). Cette étape permet de créer un fichier qui représente de manière complète le réseau routier défini dans les fichiers XML individuels.

```

C:\Users\allou\OneDrive\Bureau\python>netconvert --node-files myNodes.nod.xml --edge-files myEdges.edg.xml -t myType.type.xml -o myNet.net.xml

```

FIGURE 4.5 – Combinaison des fichiers XML avec netconvert

### 4. Génération du Trafic Routier :

- **Définition des types de véhicules et de leurs caractéristiques** : Dans un fichier de routes (.rou.xml), spécifiez les types de véhicules qui seront présents dans la simulation ainsi que leurs caractéristiques telles que la vitesse maximale, l'accélération, etc.

- **Spécification des itinéraires et des heures de départ** : Pour chaque véhicule, définissez l'itinéraire qu'il suivra dans la simulation ainsi que l'heure à laquelle il commencera son trajet.

```

<routes>
  <vType accel="1.0" decel="5.0" id="Car" length="2.0" maxSpeed="100.0" sigma="0.0" />
  <vType accel="1.0" decel="5.0" id="Bus" length="12.0" maxSpeed="1.0" sigma="0.0" />
  <route id="route0" edges="in 2to3"/>
  <vehicle depart="10" id="veh0" route="route0" type="Bus" />
  <route id="route1" edges="2to3 3to4"/>
  <vehicle depart="10" id="veh1" route="route1" type="Car" />
  <route id="route2" edges="3to4 out"/>
  <vehicle depart="30" id="veh2" route="route2" type="Car" />
</routes>

```

FIGURE 4.6 – Génération du Trafic Routier

#### 5. Configuration de SUMO :

- **Création d'un fichier de configuration SUMO (.sumocfg) :** Ce fichier permet de spécifier les fichiers de réseau et de trafic utilisés pour la simulation, ainsi que la durée de la simulation. Il permet également de définir d'autres paramètres de la simulation, tels que la période de temps simulée.

```
<configuration>
  <input>
    <net-file value="myNet.net.xml"/>
    <route-files value="myRoute.rou.xml"/>
  </input>
</configuration>
```

FIGURE 4.7 – Génération du Trafic Routier

## 4.4 Présentation de TraCI

TraCI (Traffic Control Interface) permet d'accéder à une simulation de trafic en temps réel, de récupérer les valeurs des objets simulés et de modifier leur comportement en ligne. Utilisant une architecture client/serveur, TraCI est compatible avec plusieurs langages de programmation tels que Python, C++, JAVA, et MATLAB. Elle peut aussi être combinée avec des simulateurs de réseaux de communication comme Veins ou Omnet++.

## 4.5 Implémentation du contrôle MetMod

Dans cette section, nous présenterons les résultats de simulation de la stratégie de contrôle des feux de signalisation MetMod avec 4 phases , en comparaison avec stratégie de contrôle des feux de signalisation MetMod avec 2 [30] phases et la stratégie Cournot [24] .

Les contrôles sont testés sur un réseau de quatre intersections contrôlées par des feux de signalisation, comme montré dans la figure[Fig. 4.8] :

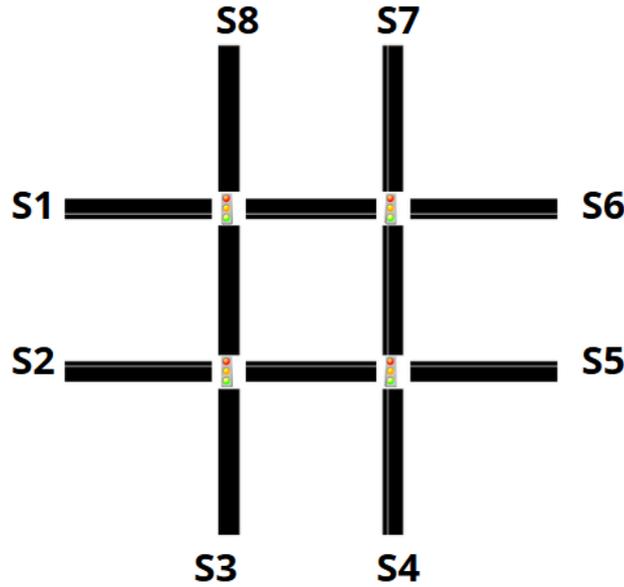


FIGURE 4.8 – Réseau de quatre intersections contrôlées par des feux de signalisation

Chaque intersection est constituées de 12 mouvements (voir la Figure (2.1)), réparti en quatre phase (voir la Figure (2.2)).

Les véhicules arrivent et quittent le réseau à partir des Région :

$$\{S1, S2, S3, S4, S5, S6, S7, S8\}$$

Le nombre moyenne de véhicules entrant sur chaque Région en 3 périodes de la durée de simulation est représenter dans le tableau [Tab. 4.1]

Région	Période 1	Période 2	Période 3
S1	390	900	600
S2	540	900	660
S3	510	660	630
S4	330	480	450
S5	420	840	660
S6	300	600	570
S7	420	570	510
S8	765	900	600

TABLE 4.1 – Flux de véhicules à l'intersection par période

Les algorithmes, ( MetMod à 2-phases, MetMod à 4-phases et Cournot ), sont évalués au cours d'une simulation d'une durée totale de 26 000 secondes, en appliquant les paramètres détaillés dans le tableau ci-dessous :

C	$g_{min,1}$	$p_1$	$g_{min,2}$	$p_2$
70s	15s	0.5	15s	0.5

TABLE 4.2 – Paramètres de simulation de la stratégie MetMod à 2-phases

C	$g_{min,1}$	$p_1$	$g_{min,2}$	$p_2$	$g_{min,3}$	$p_3$	$g_{min,4}$	$p_4$
70s	15s	0.5	15s	0.5	15s	0.5	15s	0.5

TABLE 4.3 – Paramètres de simulation de la stratégie MetMod à 4-phases

### 4.5.1 Interaction entre les Véhicules et les Contrôleurs d'Intersection

Pendant chaque cycle, les véhicules en approche d'une intersection signalisée telle que celle illustrée dans la figure [Fig. 4.8] établissent une communication avec le contrôleur de l'intersection afin de partager des données sur l'état actuel du trafic. Ainsi, dès le commencement d'un nouveau cycle, le contrôleur dispose d'informations cruciales telles que les flux d'arrivée, les flux de sortie et le nombre de véhicules en attente dans la file. Cette interaction permet une gestion plus efficace et réactive du trafic, favorisant ainsi une circulation fluide et sécurisée.

### 4.5.2 Comparaison des performances globales du réseau des stratégies MetMod 2-phase, MetMod 4-phase et Cournot.

En termes de performances globales du réseau, nous avons comparé les trois algorithmes MetMod 2-phases, MetMod 4-phases et Cournot en fonction de plusieurs critères, notamment le nombre de véhicules générés dans le réseau (Inserted vehicles), le Temps moyen de trajet ((Mean travel time), et le nombre de véhicules en circulation dans le réseau (Running vehicles).

D'après la Figure [Fig. 4.9] on a :

- **Simulation Metmod 4-phase** : 29962 véhicules insérés .
- **Simulation Metmod 2-phase** : 27609 véhicules insérés .
- **Simulation Cournot** : 22195 véhicules insérés .

D'après les résultats de la simulation utilisant l'algorithme MetMod à 4-phases a généré un nombre de véhicules insérés nettement supérieur à avec celui de l'algorithme MetMod à 2-phases, tandis que l'algorithme MetMod à deux phases a produit un nombre de véhicules insérés plus important que l'algorithme Cournot.

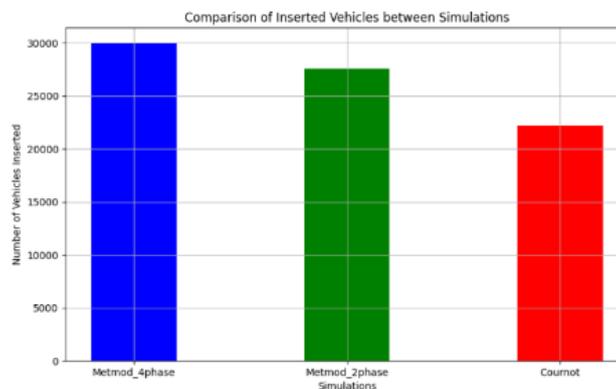


FIGURE 4.9 – Nombre de véhicules insérés dans le réseau

D'après la Figure [Fig. 4.10] on a :

- **Temps moyen de trajet de Simulation MetMod 4-phases** : 84.71 .
- **Temps moyen de trajet de Simulation MetMod 2-phases** : 107.65 .
- **Temps moyen de trajet de Simulation Cournot** : 118.68 .

D'après les résultats de la simulation, l'algorithme MetMod à 4-phases a obtenu un temps moyen de trajet nettement inférieur à celui de l'algorithme MetMod à 2-phases. De plus, l'algorithme MetMod à deux phases a enregistré un temps moyen de trajet plus court que l'algorithme Cournot.

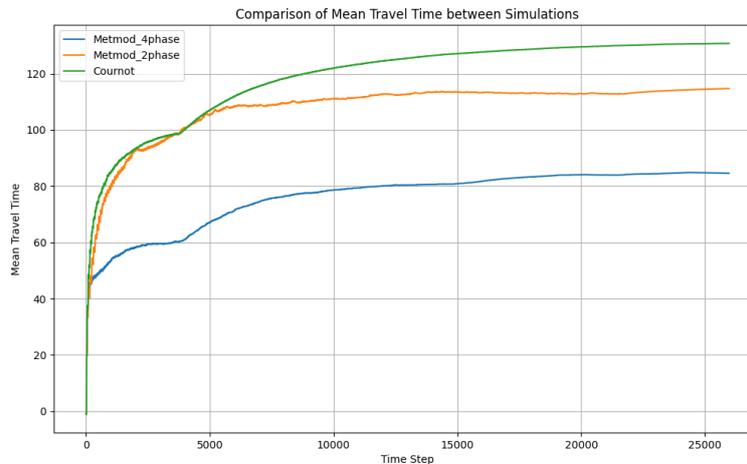


FIGURE 4.10 – Temps moyen de trajet dans le réseau

D'après la Figure [Fig. 4.11] on a :

- **Simulation Metmod-4phase** : 31 vehicles en circulation.
- **Simulation Metmod-2phase** : 45 vehicles en circulation.
- **Simulation Cournot** : 103 vehicles en circulation .

D'après les résultats de la simulation, l'algorithme MetMod à 4-phases a obtenu un nombre de véhicules en circulation nettement inférieur à celui de l'algorithme MetMod à 2-phases. De plus, l'algorithme MetMod à 2-phases a enregistré un nombre de véhicules en circulation plus faible que l'algorithme Cournot .

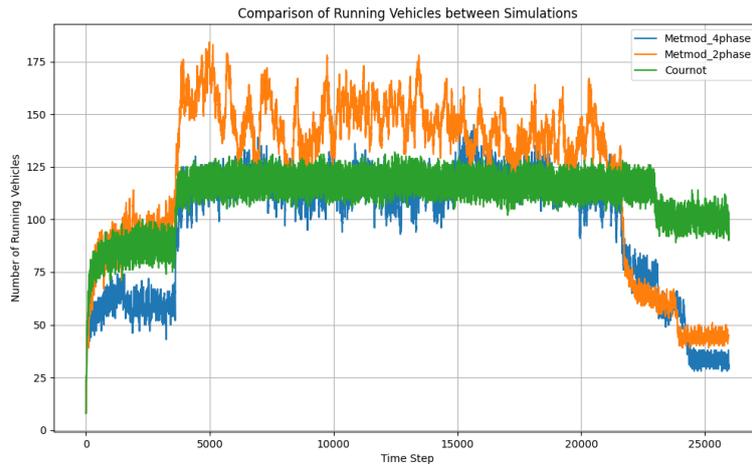


FIGURE 4.11 – Nombre de véhicules en circulation

D’après ces résultats, on conclut que l’algorithme MetMod à 4-phases semble être le plus performant parmi les trois en termes de réduction de la congestion et d’amélioration du temps de trajet moyen dans le réseau. Ensuite, l’algorithme MetMod à 2-phases présente des performances intermédiaires, tandis que l’algorithme Cournot semble être le moins efficace dans ces aspects de performance spécifiques.

## 4.6 Implémentation du contrôle MetMod-RL

Dans cette section, nous présenterons les résultats de simulation de la stratégie de contrôle des feux de signalisation MetMod-RL avec 4 phases , en comparaison avec stratégie de contrôles feux de signalisation MetMod avec 4 phases .

### 4.6.1 Paramètres du MetMod-RL

Les algorithmes, ( MetMod-RL à 4-phases et MetMod à 4-phases ), sont évalués au cours d’une simulation d’une durée totale de 26 000 secondes, en appliquant les paramètres détaillés dans le tableau ci-dessous :

C	$g_{min,1}$	$p_1$	$g_{min,2}$	$p_2$	$g_{min,3}$	$p_3$	$g_{min,4}$	$p_4$
70s	15s	0.5	15s	0.5	15s	0.5	15s	0.5

TABLE 4.4 – Paramètres de simulation de la stratégie MetMod-RL à 4-phases

Paramètre	Valeur
$\alpha$	$0.01 \leq \alpha \leq 0.1$
$\gamma$	$0.8 \leq \gamma \leq 0.95$
$\epsilon$	$0.01 \leq \epsilon \leq 0.1$
max-ep	Nombre de cycle

TABLE 4.5 – Paramètres de simulation de la stratégie MetMod-RL à 4-phases

OÙ :

Nombre de cycles = La durée Totale de la Simulation / La durée de Cycle (C)

### 4.6.2 Comparaison des performances globales du réseau des stratégies MetMod-RL 4-phase et MetMod 4-phase

En termes de performances globales du réseau, nous avons comparé les deux algorithmes MetMod-RL 4-phases et MetMod 4-phases en fonction de quatre critères, notamment le nombre de véhicules générés dans le réseau (Inserted vehicles) et le Temps moyen de trajet ((Mean travel time)).

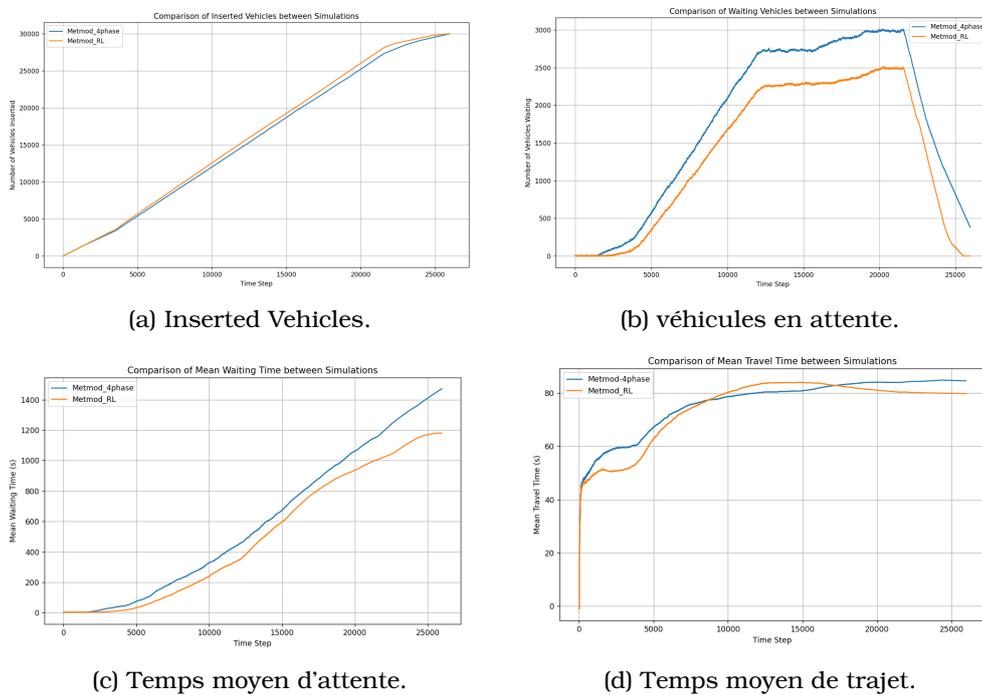


FIGURE 4.12 – Comparaison des Performances des Simulations de Contrôle MetMod-RL et MetMod-4phase.

Simulation	a	b	c	d
Metmod-4phase	29962	383	31	1471.43 s
Metmod-RL	30001	344	16	1074.9 s

TABLE 4.6 – Résultats de la Figure 4.12

### 4.6.3 Interprétation des résultats

#### 1. Nombre de véhicules insérés :

- Pour la simulation Metmod-4phase, 29962 véhicules ont été insérés, tandis que pour Metmod-RL, ce nombre est légèrement plus élevé à 30001 véhicules. Cela indique une légère

augmentation du trafic inséré dans le système pour la simulation Metmod-RL par rapport à Metmod-4phase.

## 2. Véhicules en attente :

- Le nombre de véhicules en attente est significativement plus élevé pour la simulation Metmod-4phase avec 383 véhicules, comparé à seulement 344 véhicules pour Metmod-RL. Cela suggère que la gestion des feux de signalisation dans Metmod-RL peut être plus efficace pour réduire les temps d'attente par rapport à Metmod-4phase.

## 3. Véhicules en circulation :

- La simulation Metmod-4phase montre 31 véhicules en circulation, tandis que Metmod-RL en a seulement 16. Cela pourrait indiquer une meilleure optimisation du flux de trafic et une réduction de la congestion dans Metmod-RL.

## 4. Temps moyen d'attente :

- Le temps moyen d'attente est également notablement inférieur dans la simulation Metmod-RL avec 1074.9 secondes par rapport à Metmod-4phase qui a un temps moyen d'attente de 1471.43 secondes. Cela confirme que Metmod-RL parvient à réduire efficacement le temps que les véhicules passent en attente aux intersections, améliorant ainsi la fluidité du trafic et la satisfaction des usagers.

En conclusion, la simulation Metmod-RL semble présenter des performances supérieures à Metmod-4phase en termes de gestion du trafic urbain, avec un nombre réduit de véhicules en attente, moins de véhicules en circulation, et un temps moyen d'attente significativement inférieur. Cela souligne l'efficacité des techniques ou stratégies appliquées dans Metmod-RL pour optimiser la régulation des feux de signalisation .

## 4.7 Comparaison des performances de stratégie MetMod-RL

Pour démontrer l'efficacité de notre système basé sur l'apprentissage pas renforcement , nous avons comparé ses performances dans deux scénarios : en présence de capteurs de données et en leur absence. Dans ce dernier cas, le système repose entièrement sur les prédictions fournies par les réseaux de neurones. Nos simulations, d'une durée totale de 26 000 secondes, montrent que même sans capteurs de données, les performances du système restent très proches de celles observées avec des capteurs. Les mesures clés utilisées pour cette comparaison incluent la fluidité du trafic et le temps d'attente moyen aux intersections. Les paramètres spécifiques des réseaux de neurones artificiels utilisés dans ces simulations sont détaillés comme suit :

### 4.7.1 Paramètres en cas de la présence des collecteurs de données

En cas de présence de collecteurs de données, l'algorithme MetMod-RL est évalué au cours d'une simulation d'une durée totale de 26 000 secondes, en appliquant les paramètres détaillés

dans les tableaux 4.4 et 4.5 .

### 4.7.2 Paramètres en cas de l'absence des collecteurs de données

En cas de l'absence de collecteurs de données, l'algorithme MetMod-RL et évalué au cours d'une simulation d'une durée totale de 26 000 secondes, en appliquant les paramètres détaillés dans les tableaux 4.4 et :

Paramètre	Valeur
Nombre de couches d'entrée	4
Nombre de neurones d'une couche cachée	173
Fonction d'activation	RELU
Algorithme d'optimisation	RMSprop
$\alpha$	0.001
Nombre d'époques	200

TABLE 4.7 – Paramètres de simulation de la stratégie MetMod-RL à 4-phases

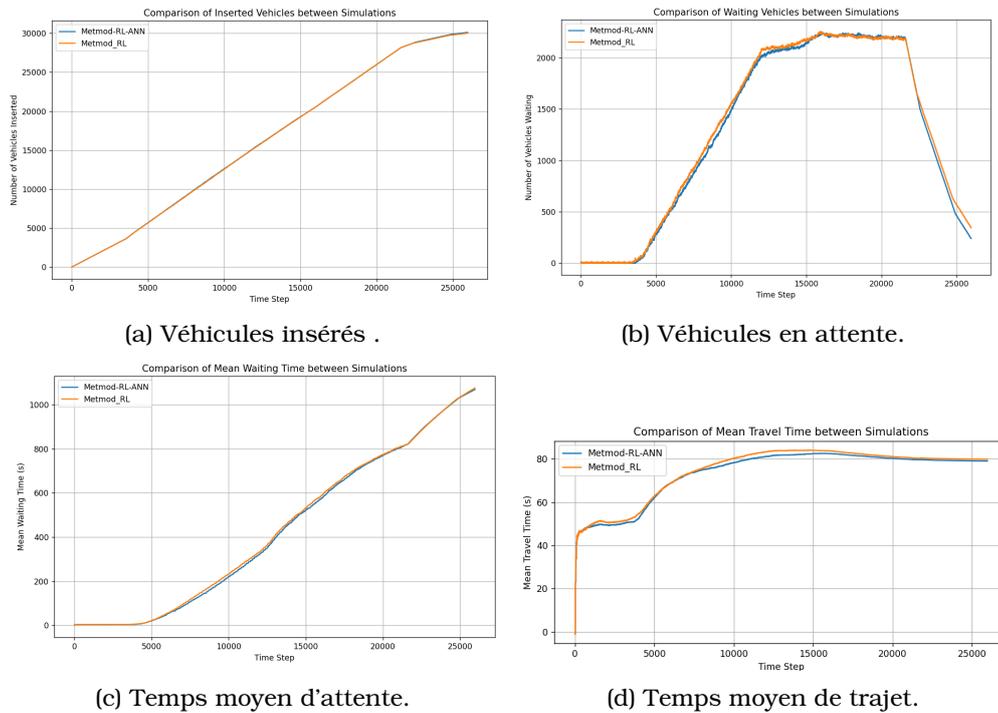


FIGURE 4.13 – Comparaison des Performances des Simulations de Contrôle MetMod-RL-ANN et MetMod-RL.

### 4.7.3 Interprétation des résultats

L'objectif principal était de déterminer si l'utilisation de réseaux de neurones pour prédire l'état du trafic lorsque les données de capteurs ne sont pas disponibles peut se rapprocher des performances de MetMod-RL utilisant des données en temps réel.

Les résultats montrent que MetMod-RL avec réseaux de neurones parvient à générer des

décisions adaptatives qui sont comparables à celles de MetMod-RL standard dans la plupart des scénarios simulés. Voici les principales observations d'après la Figure 4.13 :

- **Performance Globale** : Dans les conditions où les données de capteurs sont disponibles, les deux versions de MetMod-RL atteignent des performances similaires en termes de réduction du nombre de véhicule inséré, Véhicules en attente, Véhicules en circulation et Temps moyen d'attente .

- **Adaptabilité aux Changements** : Lorsque les capteurs sont défaillants ou absents, MetMod-RL avec réseaux de neurones montre une capacité raisonnable à ajuster les durées des feux de signalisation en fonction des prédictions de trafic. Bien que légèrement plus efficace que la version standard, cette approche parvient toujours à maintenir des performances acceptables.

En conclusion, l'intégration de réseaux de neurones dans MetMod-RL pour prédire l'état du trafic en l'absence de données de capteurs représente une solution prometteuse pour les environnements où les capteurs sont intermittents ou non disponibles. Bien que nécessitant une validation continue et des ajustements pour améliorer la précision des prédictions, cette approche offre une alternative viable et robuste pour la gestion adaptative des feux de signalisation urbains.

## Conclusion

Ce chapitre illustre l'importance croissante des techniques avancées telles que les réseaux de neurones dans l'amélioration des systèmes de contrôle de la circulation urbaine. Les résultats obtenus fournissent des bases solides pour la mise en œuvre pratique de ces solutions, en visant une gestion plus efficace et durable des flux de véhicules dans les environnements urbains .

Ce chapitre explore comment les approches d'intelligence artificielle (IA) peuvent révolutionner la gestion des feux de signalisation urbains. En utilisant des techniques avancées telles que le deep learning et l'apprentissage par renforcement, il est possible de créer des systèmes de contrôle adaptatifs et intelligents. Ces systèmes sont capables de répondre en temps réel aux fluctuations du trafic, améliorant ainsi la fluidité et la sécurité sur nos routes. L'intégration de l'IA dans la gestion des feux de signalisation ouvre la voie à des solutions plus efficaces et dynamiques pour les défis du trafic urbain moderne.

# Conclusion Générale

Ce mémoire a exploré en profondeur les défis et les innovations dans le domaine du contrôle de la circulation urbaine, en se concentrant sur l'analyse comparative de différentes stratégies et l'introduction de techniques avancées telles que l'apprentissage par renforcement.

- Dans le premier chapitre, nous avons posé les bases en décrivant l'importance cruciale d'une gestion efficace du trafic urbain pour améliorer la sécurité routière, réduire la congestion et minimiser l'empreinte carbone. Nous avons souligné l'évolution des défis urbains modernes et la nécessité croissante de solutions intelligentes et adaptatives.

- Dans le deuxième chapitre, nous explorons l'optimisation du contrôle des feux de signalisation à une intersection en utilisant des approches mathématiques. Les mouvements à l'intersection sont structurés en phases, chaque phase étant contrôlée comme un joueur dans un jeu non coopératif. L'objectif principal est de minimiser la congestion en déterminant la durée du feu vert pour chaque phase, tout en respectant des contraintes de capacité et de temps. Le méta-jeu ainsi formulé cherche un équilibre de Nash généralisé (GNE) comme solution d'équilibre, dont le calcul est ramené à la résolution d'un programme linéaire et des conditions de KKT.

- Le troisième chapitre explore les méthodes avancées d'intelligence artificielle pour améliorer la gestion des feux de signalisation urbains. En intégrant l'apprentissage supervisé, non supervisé et par renforcement, ainsi que l'utilisation de réseaux de neurones artificiels, ces approches visent à rendre les systèmes de signalisation plus adaptatifs et efficaces. L'approche MetMod-RL, combinant la théorie des jeux et l'apprentissage par renforcement, est particulièrement prometteuse pour minimiser la congestion et optimiser la fluidité du trafic, même en l'absence de données directes de capteurs. Ces avancées ouvrent la voie à des systèmes urbains plus intelligents et écologiquement durables.

- Le dernier chapitre, se concentre sur l'évaluation comparative des stratégies de contrôle des feux de signalisation urbains. Il analyse principalement les performances des stratégies MetMod à 2-phases, MetMod à 4-phases et l'algorithme Cournot à travers des simulations détaillées.

Les résultats indiquent que l'algorithme MetMod à 4-phases se distingue par une gestion plus efficace du trafic urbain, avec une réduction significative du nombre de véhicules insérés, des temps d'attente aux intersections et du temps moyen de trajet par rapport aux autres stratégies. L'intégration de l'algorithme MetMod-RL, qui combine l'apprentissage par renforcement avec des réseaux de neurones, améliore encore ces performances en adaptant dynamiquement les temps

des feux de signalisation en fonction des prédictions de trafic.

Cette étude contribue ainsi à enrichir la compréhension des systèmes de gestion de trafic urbain et propose des solutions innovantes pour répondre aux défis croissants de la mobilité urbaine. Les résultats obtenus offrent des perspectives prometteuses pour l'application pratique de ces technologies avancées dans les environnements urbains modernes.

# Bibliographie

- [1] Agrawal, A., & Paulus, R. (2020, janvier). Intelligent traffic light design and control in smart cities : a survey on techniques and methodologies. *International Journal of Vehicle Information and Communication Systems*. DOI : 10.1504/UVICS2020.111456.
- [2] Baudin, L. (2023). Contributions on Online Learning in Stochastic Games. PhD thesis, Université Paris sciences et lettres.
- [3] bai2021general, A general framework for intersection traffic control with backpressure routing, Bai, Sen and Bai, Xin, IEEE Access, 9, 102125–102136, 2021, IEEE.
- [4] Belakbir, N. (2011). Application de la théorie des jeux dans les réseaux de télécommunications, Mémoire de magistère en mathématiques appliquées. Université abderrahmane mira de béjaia.
- [5] Benyamine, M. (2020). Deep Reinforcement Learning for Complex Manipulation Tasks with Sparse Feedback. A thesis submitted in fulfillment of the requirements for the degree of Master of Industrial Engineering and Management, Ben-Gurion University of the Negev.
- [6] Bibi Mohand Ouamer (2022, 2023). cours d'optimisation global
- [7] burkov2007adaptive, Adaptive Dynamics Learning and Q-initialization in the context of multiagent learning, Burkov, Andriy, 2007
- [8] Braban, C., & Boillot, F. (2003). LES SYSTEMES TEMPS REEL DE COMMANDE DE FEUX EN MILIEU URBAIN [REAL TIME URBAN TRAFFIC SIGNAL CONTROL SYSTEMS].
- [9] Bretherton, R. D. (1990). SCOOT Urban Traffic Control System—Philosophy and Evaluation. In *Control, Computers, Communications in Transportation* (pp. 237-239). Pergamon
- [10] Bwachs, M., *Des outils de gestion du trafic et des feux de signalisation en particulier*.
- [11] Chen, C., Wei, H., Xu, N., Zheng, G., Yang, M., Xiong, Y., Xu, K., & Li, Z. (2020). Toward a thousand lights : Decentralized deep reinforcement learning for large-scale traffic signal control. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 34, No. 04, pp. 3414–3421).
- [12] Costa, J. M. *Système de transports intelligents*. UIT. Canada : 2021, 6-11.

- 
- [13] Gao, K., Zhang, Y., Zhang, Y., Su, R., Suganthan, P. N. (2018). Meta-heuristics for bi-objective urban traffic light scheduling problems. *IEEE Transactions on Intelligent Transportation Systems*, 20(7), 2618–2629. IEEE.
- [14] Ghazal, B., ElKhatib, K., Chahine, K., Kherfan, M. (2016, April). "Smart traffic light control system." In 2016 third international conference on electrical, electronics, computer engineering and their applications (EECEEA) (pp. 140-145). IEEE.
- [15] Hamilton, A., Waterson, B., Cherrett, T., Robinson, A., & Snell, I. (2013). *The evolution of urban traffic control : changing policy and technology*. *Transportation planning and technology*, 36(1), 24-43.
- [16] Hong, W., Tao, G., Wang, H., Wang, C. (2022). Traffic signal control with adaptive online-learning scheme using multiple-model neural networks. *IEEE Transactions on Neural Networks and Learning Systems*. IEEE.
- [17] J Ben Rosen. Existence and uniqueness of equilibrium points for concave n-person games. *Econometrica : Journal of the Econometric Society*, pages 520–534, 1965.
- [18] Jamal, A., Al-Ahmadi, H. M., Butt, F. M., Iqbal, M., Almoshaogeh, M., Ali, S. (2021). *Meta-heuristics for Traffic Control and Optimization : Current Challenges and Prospects*. IntechOpen.
- [19] Jagannatham, A. K., & Kumar, V. (2016). Introduction to game theory. In *Decision Sciences* (pp. 95–160). CRC Press.
- [20] Jean-Pierre Aubin. *Optima and equilibria : an introduction to nonlinear analysis*, volume 140. Springer Science Business Media, 2013.
- [21] Kamesh, D., Sumadhuri, D., Sahithi, M., Sastry, J. (2017). An efficient architectural model for building cognitive expert system related to traffic management in smart cities.
- [22] Kazaryan, D. E., Mihalyev, V. A., Sofronova, E. A. (2017). Artificial neural network approach to traffic flow control. *RUDN Journal of Engineering Research*, 18(1), 97–106.
- [23] Khac-Hoai Nam Bui, Jai E Jung, and David Camacho. Game theoretic approach on real-time decision making for IoT-based traffic light control. *Concurrency and Computation : Practice and Experience*, 29(11), e4077, 2017.
- [24] Khac-Hoai Nam Bui, Jai E Jung, and David Camacho. Game theoretic approach on realtime decision making for iot-based traffic light control. *Concurrency and Computation : Practice and Experience*, 29(11) :e4077, 2017
- [25] Krajzewicz, Daniel, Jakob Erdmann, Michael Behrisch, and Laura Bieker. "Recent development and applications of SUMO-Simulation of Urban MObility." *International journal on advances in systems and measurements* 5, no. 3&4 (2012) : Citeseer.
- [26] Kumar, R., Sharma, N. V. K., & Chaurasiya, V. K. (2024). Adaptive traffic light control using deep reinforcement learning technique. *Multimedia Tools and Applications*, 83(5), 13851–13872. Springer.
-

- 
- [27] Liang, X., Du, X., Wang, G., & Han, Z. (2018). Deep reinforcement learning for traffic light control in vehicular networks. *arXiv preprint arXiv :1803.11115*.
- [28] Liang, X., Du, X., Wang, G., & Han, Z. (2019). A deep reinforcement learning network for traffic light cycle control. *IEEE Transactions on Vehicular Technology*, 68(2), 1243–1253. IEEE.
- [29] López, S., Hernández, P., Hernández, A., García, M. (1999). Artificial neural networks as useful tools for the optimization of the relative offset between two consecutive sets of traffic lights. In *International Work-Conference on Artificial Neural Networks* (pp. 795–804). Springer.
- [30] Mahiout, H. (2023). Stratégies innovantes de régulation du trafic urbain (Thèse de doctorat en Mathématiques appliquées, Université de Bejaia). Soutenue le 14 juin 2023 sous la direction du Pr. Radjef Mouhamed Said et Farhi Nadir.
- [31] matsuo2022deep. Matsuo, Y., LeCun, Y., Sahani, M., Precup, D., Silver, D., Sugiyama, M., Uchibe, E., Morimoto, J. (2022). learning, reinforcement learning, and world models. *Neural Networks*, 152, 267–275. Elsevier.
- [32] Ministère des Transports, Mohamed Seddik, et Brahim Jijel, "Développement d'un système intelligent de gestion des feux de circulation à Base des algorithmes génétiques Encadré", 2020.
- [33] Mohandas, P., Dhanaraj, J. S. A., Gao, X.-Z. (2019). Artificial neural network based smart and energy efficient street lighting system : A case study for residential area in Hosur. *Sustainable Cities and Society*, 48, 101499. Elsevier.
- [34] Navarro-Espinoza, A., López-Bonilla, O. R., García-Guerrero, E. E., Tlelo-Cuautle, E., López-Mancilla, D., Hernández-Mejía, C., & Inzunza-González, E. (2022). Traffic flow prediction for smart traffic lights using machine learning algorithms. *Technologies*, 10(1), 5. MDPI.
- [35] Nwiabu, N. D., Udoudom, E. E. (2018). Traffic Light Control System using Genetic Algorithm. *International Journal of Computer Applications*, 975, 8887.
- [36] parizeau2004reseaux, Réseaux de neurones, Parizeau, Marc, GIF-21140 et GIF-64326, 124, 2004
- [37] RADJEF, M. S. (2022, 2023). Cours de la théorie des jeux.
- [38] Savithramma, R. M., Sumathi, R., & Sudhira, H. S. (2022). A comparative analysis of machine learning algorithms in design process of adaptive traffic signal control System. In *Journal of Physics : Conference Series* (Vol. 2161, No. 1, p. 012054). IOP Publishing.
- [39] Sharma, M., Bansal, A., Kashyap, V., Goyal, P., & Sheikh, T. H. (2021). Intelligent traffic light control system based on traffic environment using deep learning. In *IOP Conference Series : Materials Science and Engineering* (Vol. 1022, No. 1, p. 012122). IOP Publishing.
- [40] sutton2018reinforcement Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning : An introduction (2nd ed.). *MIT Press*.
- [41] sutton1998reinforcement, The reinforcement learning problem, Sutton, Richard S and Barto, Andrew G, Reinforcement learning : An introduction, 51–85, 1998, MIT press
-

- [42] Tan Linglong, Zhao Xiaohua, Hu Dunli, Shang Yanzhang, and Wan Ren. A study of single intersection traffic signal control based on two-player cooperation game model. In *2010 WASE International Conference on Information Engineering*, volume 2, pages 322–327. IEEE, 2010.
- [43] THAI-PHU NGUYEN, *Conception et application d'un modèle de l'information routière et ses effets sur le trafic*, Thèse de doctorat, Université Paris-Est, 2010. Disponible sur : <https://pastel.archives-ouvertes.fr/tel-00626631/document>.
- [44] Tlig, M., *Coordination locale et optimisation distribuée du trafic de véhicules autonomes dans un réseau routier*, Thèse de doctorat, 2015.
- [45] Vehicle Actuated Signal. Disponible sur : [www.indiamart.com/proddetail/vehicle-actuated-signal-18477376812.html](http://www.indiamart.com/proddetail/vehicle-actuated-signal-18477376812.html). Consulté le 05/04/2020 à 21 :28.
- [46] Vidali, A., Crociani, L., Vizzari, G., Bandini, S. (2019). A Deep Reinforcement Learning Approach to Adaptive Traffic Lights Management. University of Milano-Bicocca, Milano, Italy.
- [47] Wei, H., Zheng, G., Gayah, V., & Li, Z. (2021). Recent advances in reinforcement learning for traffic signal control : A survey of models and evaluation. *ACM SIGKDD Explorations Newsletter*, 22(2), 12–18. ACM New York, NY, USA.
- [48] Xing, J., Chu, L., Guo, C. (2021). Optimization of energy consumption based on traffic light constraints and dynamic programming. *Electronics*, 10(18), 2295. MDPI.
- [49] Xing, J., Chu, L., Guo, C. (2021). Optimization of energy consumption based on traffic light constraints and dynamic programming. *Electronics*, 10(18), 2295. MDPI.
- [50] Y.Zennir, *Apprentissages par renforcement et systèmes distribués : Application à l'apprentissage de la marche d'un robot hexapode*, Thèse de doctorat, E.E.A, Lyon , 2004.
- [51] Yin, B., Dridi, M., El Moudni, A. (2015). Forward search algorithm based on dynamic programming for real-time adaptive traffic signal control. *IET Intelligent Transport Systems*, 9(7), 754–764. Wiley Online Library.
- [52] Zhang, W., Ma, Y., Wang, Y., & Lv, X. (2021). *A Review of Intelligent Traffic Signal Control Systems : Principles, Techniques, and Applications*. IEEE Access, 9, 106444-106463.
- [53] Zantalis, F., Koulouras, S., & Kandris, D. (2019). A Review of Machine Learning and IoT in Smart Transportation. *Future Internet*, 11, 94. DOI : 10.3390/fi11040094.
- [54] Zang, X., Yao, H., Zheng, G., Xu, N., Xu, K., & Li, Z. (2020). Metalight : Value-based meta-reinforcement learning for traffic signal control. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 34, No. 01, pp. 1153–1160).

---

**Résumé :**

Ce mémoire explore en profondeur les défis et les innovations du contrôle de la circulation urbaine, en se concentrant sur l'analyse comparative des stratégies et l'intégration de techniques avancées comme l'apprentissage par renforcement. Il souligne l'importance cruciale d'une gestion efficace du trafic pour la sécurité routière. À travers des approches mathématiques et l'utilisation d'intelligence artificielle, notamment l'apprentissage automatique et par renforcement, l'étude vise à rendre les systèmes de signalisation urbains plus adaptatifs et efficaces. Les résultats montrent que l'approche MetMod-RL, combinant théorie des jeux et apprentissage par renforcement, offre des performances significatives dans l'optimisation du trafic urbain. En conclusion, ce mémoire contribue à promouvoir des solutions urbaines intelligentes et durables, améliorant la mobilité urbaine.

---

**Mots clés :** *Contrôle de la circulation urbaine, Feux de signalisation, Optimisation des intersections, Apprentissage par renforcement, Réseaux de neurones, la théorie des jeux, Simulation de trafic, Modélisation du trafic urbain, Réduction de la congestion .*

---

**Abstract :**

In this work, we explore the challenges and innovations in urban traffic control, focusing on comparative analysis of strategies and integration of advanced techniques such as reinforcement learning. It emphasizes the crucial importance of effective traffic management for road safety. Through mathematical approaches and the use of artificial intelligence, including machine learning and reinforcement learning, the study aims to make urban signaling systems more adaptive and efficient. The results demonstrate that the MetMod-RL approach, combining game theory and reinforcement learning, delivers significant performance improvements in urban traffic optimization. In conclusion, this thesis contributes to promoting intelligent and sustainable urban solutions, enhancing urban mobility.

---

**Keywords :** *Urban traffic control, Traffic lights, Intersection optimization, Reinforcement learning, Neural networks, Game theory, Traffic simulation, Urban traffic modeling, Congestion reduction*

---