

Université MUSTAPHA Stambouli

Mascara



جامعة مصطفى اسطمبولي

معسكر

Faculté des Sciences Exactes

Département d'Informatique

## THESE de DOCTORAT

Spécialité Modélisation et Optimisation des Systèmes

Intitulée

Application des Méta-heuristiques pour la Commande des Feux de  
Signalisation dans un Réseau de Plusieurs Intersections

*Présentée par* : Kouidri Chaima

Le 04/12/2024

Devant le jury :

Président	Salem Mohammed	Pr	Université de Mascara
Examineur	Hamou Reda Mohamed	Pr	Université de Saida
Examineur	Houari Amina	MCA	Université de Mascara
Encadreur	Bachir Bouiadjra Rochdi	MCA	Université de Mascara
Co-Encadreur	Mahi Faiza	MCB	Université de Mascara

Année Universitaire : 2024-2025



سبحانك لا علم لنا إلا ما علمتنا

إنك أنت العليم الحكيم



سورة البقرة: الآية: 31

---

# DÉDICACE

A la mémoire de mon père,  
A ma tendre grande mère,  
A ma mère a qui je dois tout mon bonheur  
pour son soutien sans faille et sa confiance,  
A mes chers frères et sœurs,  
A tous mes amis,  
A tous ceux qui me sont chers.

# REMERCIEMENTS

Nous remercions ALLAH de nous avoir donné le courage et la volonté ainsi que la conscience et la patience d'avoir pu terminer notre thèse de doctorat.

Je tiens à exprimer mes profondes gratitude à Monsieur Rochdi Bachir Bouiadjra, Maître de conférence -A- à l'université de Mascara Mustapha Stambouli, pour avoir encadré et dirigé mes recherches. Je le remercie pour m'avoir soutenu et appuyé tout au long de ma thèse de doctorat. Ses précieux conseils, son exigence et ses commentaires ont permis d'améliorer grandement la qualité de mes travaux.

Ma gratitude sincère et profonde est adressée à mon co-encadreur Madame Mahi Faiza, Maître de conférence -B- à l'université de Mascara Mustapha Stambouli, pour m'avoir co-encadrée et soutenue durant ces années de thèse. Je la remercie aussi pour son aide précieuse, les conseils et les connaissances dont elle a su me faire profiter.

Je voudrais, également, exprimer mes sentiments de gratitude à tous les membres du jury qui ont accepté d'évaluer notre travail tout en espérant qu'ils trouvent dans ce rapport la clarté et la motivation qu'ils attendent.

Un grand MERCI à tous mes professeurs depuis l'école élémentaire, à tous ces forgerons du savoir et de la connaissance.

Ma gratitude et mes remerciements à ma famille. Merci à ma mère et ma grande mère de m'avoir donné envie d'apprendre, de persévérer, pour leurs sacrifices, pour leurs encouragements et soutiens sans faille. Mes rêves sont un peu les vôtres...

Mes remerciements vont aussi à tous ceux et celles qui ont participé de près ou de loin à l'élaboration du présent travail, à tous nos amis et collègues pour leur soutien moral tout au long de la préparation de ce travail.

## ملخص

يلعب النقل دورًا حاسمًا في التنمية الاقتصادية حيث يمكن لنظام نقل الفعال أن يعزز النمو الاقتصادي ويحسن الرفاه الاجتماعي. فشركات الطرق من بين الأنظمة الحيوية الأكثر أهمية التي تشهد ازدحامًا شديدًا بسبب الاضطرابات الرئيسية ، بسبب هذه الاضطرابات تتراجع كفاءة نظام المرور وتزداد الازدحامات في شبكة الطرق.

تقترح هذه الأطروحة نهجًا جديدًا لتحسين إشارات المرور بهدف تقليل الازدحام المروري وتحسين سلاسة الحركة.

لتحقيق هذا الهدف ، يتم استخدام اثنتين من الاستدلالات الفوقية : الخوارزميات الجينية ، خوارزمية بحث الكائنات الحية التكافلية.

تتضمن الأطروحة دراسة معمقة لهذين النهجين وتقترح تجارب وتحليلات لتقييم أدائهما في سيناريوهات مرور مختلفة. تظهر نتائج الأطروحة أن الاستدلالات الفوقية المقترحة قادرتان على تحسين إشارات المرور بفعالية.

**الكلمات المفتاحية** : الخوارزميات الجينية ، خوارزمية بحث الكائنات الحية التكافلية ، الاستدلالات الفوقية ، إشارات المرور ، النقل ، شبكات الطرق.

---

# ABSTRACT

Transportation plays a critical role in economic development, an efficient transportation system can promote economic growth and enhance social well-being. Road networks, as part of a transportation system, are among the most important vital systems. Urban road networks experience high congestion due to infrequent major disruptions. As a result of these disruptions, the performance of the traffic system is reduced, and congestion in the road network is increased. This thesis proposes an innovative approach for optimizing traffic signal lights with the aim of reducing road congestion and improving traffic flow. To achieve this goal, two metaheuristics are utilized : genetic algorithms (GA) and discrete symbiotic organism search (DSOS).

The thesis includes a comprehensive study of these two approaches and proposes experiments and analyses to evaluate their performance in different traffic scenarios. The results of the thesis demonstrate that both metaheuristics are capable of effectively optimizing traffic signal lights. However, significant differences are observed in terms of performance and traffic volume.

**Keywords** : Transport, Road Networks, Meta-heuristics , Genetic Algorithm(GA) , Discret Symbiotic Organism Search(DSOS) , Traffic lights.

---

# RÉSUMÉ

Le transport a un rôle critique dans le développement économique, un système de transport efficace peut favoriser la croissance économique et améliorer le bien-être social. Les réseaux routiers, en tant que partie d'un système de transport, sont parmi les systèmes vitaux les plus importants. Les réseaux routiers urbains connaissent une forte congestion en raison de perturbations majeures. En raison de ces perturbations, les performances du système de circulation sont réduites et la congestion dans le réseau routier sont augmentés.

Cette thèse propose une approche novatrice pour l'optimisation des feux de signalisation dans le but de réduire la congestion routière et d'améliorer la fluidité du trafic. Pour atteindre cet objectif, deux méta-heuristiques sont utilisées : les algorithmes génétiques (AG) et la recherche symbiotique d'organismes discrets (DSOS).

La thèse comprend une étude approfondie de ces deux approches et propose des expériences et des analyses pour évaluer leur performance dans différents scénarios de circulation. Les résultats de la thèse montrent que les deux méta-heuristiques sont capables d'optimiser efficacement les feux de signalisation.

**Mots clés :** Transport , Réseau routiers , Méta-heuristiques , Algorithme Génétique(AG) , recherche d'organisme Symbiotique Discret (DSOS) , Feux de signalisation.

# Table des matières

<b>Introduction Générale</b>	<b>12</b>
<b>1 Méthodes de modélisation et de contrôle du trafic</b>	<b>17</b>
1.1 Introduction . . . . .	19
1.2 Historique . . . . .	19
1.3 Modélisation et contrôle de trafic . . . . .	20
1.3.1 Paramètre de transport . . . . .	20
1.3.2 Modèles de trafic . . . . .	24
1.4 Système de transport intelligent . . . . .	27
1.4.1 Communications sans fil . . . . .	28
1.4.2 Système De Positionnement Global (GPS) . . . . .	28
1.4.3 Véhicule autonome . . . . .	29
1.4.4 Communications V2V et V2I . . . . .	30
1.4.5 Capteur vidéo . . . . .	31
1.4.6 Capteurs micro-ondes . . . . .	31
1.4.7 Communications à courte portée . . . . .	32
1.4.8 Détection de boucle inductive . . . . .	32
1.5 Contre-mesures pour la congestion du trafic . . . . .	32
1.5.1 Transport public . . . . .	32
1.5.2 Planification de l'itinéraire . . . . .	33
1.5.3 Contrôle des feux de circulation . . . . .	33
1.5.4 Prédiction du trafic . . . . .	33
1.5.5 Stratégies de tarification . . . . .	33
1.5.6 Infrastructures routières . . . . .	34
1.6 Conclusion . . . . .	34
<b>2 Méthodes Bio-Inspirées</b>	<b>35</b>
2.1 Introduction . . . . .	37
2.2 Techniques d'optimisation des métaheuristiques . . . . .	38



## TABLE DES MATIÈRES

---

2.3	Algorithme d'optimisation des colonies des fourmis (ACO) . . . . .	40
2.3.1	Principe . . . . .	40
2.3.2	Fonctionnement d'Algorithme . . . . .	41
2.3.3	Travaux connexes . . . . .	42
2.4	Algorithme d'optimisation par essais de particules (PSO) . . . . .	43
2.4.1	Principe . . . . .	43
2.4.2	Fonctionnement d'Algorithme . . . . .	44
2.4.3	Travaux connexes . . . . .	45
2.5	Colonie d'abeilles artificielles (ABC) . . . . .	46
2.5.1	Principe . . . . .	46
2.5.2	Fonctionnement d'algorithme . . . . .	46
2.5.3	Travaux connexes . . . . .	46
2.6	Recherche de coucou (CS) . . . . .	47
2.6.1	Principe . . . . .	47
2.6.2	Fonctionnement . . . . .	48
2.6.3	Travaux connexes . . . . .	49
2.7	Algorithme de chauve-souris (Bat algorithm (BA)) . . . . .	50
2.7.1	Principe . . . . .	50
2.7.2	Fonctionnement . . . . .	50
2.7.3	Travaux connexes . . . . .	51
2.8	Système immunitaire artificiel (AIS) . . . . .	51
2.8.1	Principe . . . . .	51
2.8.2	Fonctionnement . . . . .	52
2.8.3	Travaux connexes . . . . .	52
2.9	Optimiseur des loups gris (GWO) . . . . .	53
2.9.1	Principe . . . . .	53
2.9.2	Fonctionnement . . . . .	54
2.9.3	Travaux connexes . . . . .	55
2.10	Recherche d'harmonie (HS) . . . . .	55
2.10.1	Principe . . . . .	55
2.10.2	Fonctionnement . . . . .	56
2.10.3	Travaux connexes . . . . .	56
2.11	Conclusion . . . . .	56
<b>3</b>	<b>Approches Proposées</b> . . . . .	<b>58</b>
3.1	Introduction . . . . .	59
3.2	Optimisation et méthodes méta-heuristiques . . . . .	60
3.2.1	Définition de l'optimisation . . . . .	60
3.2.2	Problème d'optimisation . . . . .	61
3.2.3	Méthodes classiques v.s. Méthodes métaheuristiques . . . . .	62
3.2.3.1	Caractéristiques des métaheuristiques . . . . .	63

## TABLE DES MATIÈRES

---

3.2.3.2	Classification des métaheuristiques . . . . .	64
3.3	Approche 1 : Description de la première stratégie proposée (DJ-GA)	65
3.3.1	Niveau1 : Détermination des feux de signalisation critiques	67
3.3.2	Description de la fonction objective . . . . .	68
3.3.2.1	Description détaillée des paramètres de l'équation (3.7) . . . . .	69
3.3.3	Niveau 2 : Optimisation des feux de signalisation par l'Algorithme Génétique . . . . .	70
3.3.3.1	Fonctionnement des Algorithmes Génétiques . . . . .	71
3.3.4	Pourquoi les algorithmes génétiques fonctionnent-ils? . . . . .	73
3.3.4.1	Justifications qualitatives des étapes d'optimisation	73
3.3.5	Adaptation de l'Algorithme Génétique . . . . .	74
3.3.5.1	Représentation du Chromosome . . . . .	75
3.3.5.2	Processus de Sélection . . . . .	76
3.3.5.3	Processus de Croisement . . . . .	76
3.3.5.4	Processus de Mutation . . . . .	77
3.4	Approche 2 : Description de la stratégie proposée(ACO-DSOS) . . . . .	78
3.4.1	Niveau 1 : Recherche des feux critiques par l'algorithme de colonie des fourmis (ACO) . . . . .	79
3.4.1.1	Adaptation de colonie de fourmis sur la Détermination des feux de Signalisation Critiques . . . . .	80
3.4.2	Niveau02 :Optimisation des feux de signalisation par l'Algorithme Discret Symbiotic Organism Search . . . . .	82
3.4.2.1	Symbiotic Organism Search . . . . .	82
3.4.2.2	Adaptation d'Algorithme Discret Symbiotic Organism Search sur le problème des feux de signalisation . . . . .	84
3.5	Conclusion . . . . .	89
<b>4</b>	<b>Résultats Experimentaux</b>	<b>90</b>
4.1	Introduction . . . . .	91
4.2	Langage et Environnement de travail . . . . .	91
4.2.1	Langage de programmation JAVA . . . . .	91
4.2.2	Environnement de travail Netbeans . . . . .	91
4.2.3	Python . . . . .	92
4.2.4	Simulateur SUMO . . . . .	92
4.2.4.1	Principe de fonctionnement . . . . .	93
4.2.4.1.1	Applications incluses : . . . . .	94
4.3	Étude de cas : régulation du trafic de la zone d'Alger (Bir morad rais - El moradia - Hydra) . . . . .	95

## TABLE DES MATIÈRES

---

4.4	Résultats Expérimentaux . . . . .	98
4.4.1	Résultats de l'approche d'optimisation de feux basée sur les Algorithmes Génétiques . . . . .	98
4.4.1.1	Configuration de simulation . . . . .	98
4.4.1.2	résultat 1 : comparaison de l'approche statique avec l'approche DJ-GA . . . . .	99
4.4.1.3	Impact de Variation du Nombre d'Itérations . . . . .	102
4.4.1.4	Impact de Variation du Nombre de Population . . . . .	103
4.4.2	Résultats de l'approche d'optimisation de feux basée sur l'Algorithme Discret Symbiotique Organisme Search (DSOS) . . . . .	104
4.4.2.1	Simulation et résultats . . . . .	104
4.4.2.2	Niveau01 : Détermination des feux critiques . . . . .	104
4.4.2.3	Niveau02 : Application d'algorithme Discret Organism Search (DSOS) . . . . .	104
4.5	Conclusion . . . . .	108
	<b>conclusion et Perspectives</b>	<b>109</b>
	Bibliographie . . . . .	111
	Table des figures	120
	Liste des tableaux	122

# LISTE DES ACRONYMES

GPS : Global Positioning System  
ITS : Intelligent Traffic System  
WSN : Wireless Sensor Network  
RADAR : Radio Detection And Ranging  
ACO : Ant Colony Optimisation  
PSO : Particle Swarm Optimisation  
ABC : Artificial Bee Colony  
CS : Cuckoo Search  
BA : Bat Algorithm  
AIS : Artificial Immune System  
GWO : Grey Wolf Optimizer  
HS : Harmony Search  
GA : Genetic Algorithm  
DSOS : Discret Symbiotic Organism Search  
LP : Linear Programming  
DP : Dynamic Programming  
BB : Branch and Bound  
V2V : Vehicle to vehicle  
V2I : vehicle to Infrastructure  
OSPF : Open Shortest Path First  
TS : Tabou Search  
SA : Recuit Simulé  
PI : Performance Index  
SUMO : Simulation of Urban MObility

# INTRODUCTION GÉNÉRALE

## CONTEXTE

Les embouteillages deviennent un problème majeur auquel sont confrontés la plupart des pays du monde. Il y a une augmentation constante du taux de population et donc une augmentation du nombre de routes et de véhicules qui provoquent des embouteillages [1]. En conséquence, les conducteurs et les voyageurs sont confrontés à de nombreux problèmes tels que la pollution de l'air, la perte de temps, la consommation de carburant, la frustration, les pertes économiques et d'autres problèmes graves [2].

Il existe un certain nombre de solutions suggérées pour atténuer le problème. Les urbanistes ont tenté de lutter contre ce phénomène en construisant de nouvelles voies, des ponts et de les agrandir [3]. Cependant, il n'a pas rencontré le succès escompté. Le premier problème avec cette solution est chère et il est impossible de le faire dans les villes urbaines en raison des zones résidentielles et des bâtiments à proximité [4]. Les chercheurs recourent donc à l'utilisation optimale des infrastructures disponibles [5].

Dans les systèmes de circulation, il existe une relation entre le moment des feux de circulation et le temps de déplacement total pour tous les véhicules dans le réseau, de sorte que l'ajustement du moment des feux peut donner plus de temps vert à une intersection avec un trafic intense ou raccourcir ou même sauter une phase qui a peu ou pas du trafic d'attente. Ainsi, il peut conduire à augmenter ou à diminuer le temps de parcours des véhicules [6]. Lorsqu'on choisit le temps de parcours moyen comme mesure de l'efficacité du réseau de circulation, les meilleures valeurs pour le temps des feux de circulation sont celles qui donnent le temps de parcours moyen minimum pour tous les véhicules.

Le contrôle intelligent des feux de signalisation et le système de coordination sont de plus en plus nécessaires pour garantir que le trafic se déroule aussi bien que possible[1]. Ces systèmes intelligents peuvent être développés en remplaçant les systèmes de feux de circulation traditionnels par des systèmes intelligents qui s'ajustent automatiquement en fonction des données historiques collectées

par les détecteurs (capteurs, caméras).

## PROBLÉMATIQUES ET MOTIVATIONS

La programmation des feux de circulation est considérée comme un problème NP-difficile [7]. C'est un problème du monde réel où la solution optimale est inconnue. De plus en plus le nombre de feux de circulation est grand, plus l'espace de recherche du problème est grand alors la complexité de la recherche sera beaucoup plus élevée [9].

La grande majorité des problèmes d'optimisation du monde réel dans plusieurs domaines tels que le transport, l'ingénierie, la fabrication, etc sont des problèmes NP-difficiles [9]. Pour les problèmes d'optimisation complexes (par exemple, l'optimisation NP-difficile ou globale), les algorithmes exacts ne sont pas appropriés car le temps nécessaire pour trouver la solution optimale peut augmenter de façon exponentielle par rapport aux dimensions du problème [10]. Par conséquent, les méthodes heuristiques sont plus adaptées pour résoudre des problèmes complexes avec un espace de recherche de grande dimension où il a tendance à trouver une bonne solution dans un délai raisonnable [9].

Les méthodes heuristiques peuvent être classées en deux types : les heuristiques spécifiques conçues pour des problèmes spécifiques (problème dépendant) et les métaheuristiques développées pour résoudre un large éventail de problèmes (problème indépendant) ([9]; [10]).

Les algorithmes de métaheuristiques ont montré des performances supérieures dans la résolution d'une très grande variété de problèmes d'optimisation tels que les problèmes d'ordonnancement, l'optimisation des paramètres, la sélection de fonctionnalités, le regroupement automatique, la formation des réseaux de neurones ([12]; [11]). Ces algorithmes sont devenus populaires pour résoudre le problème de planification des feux de circulation ([13]; [2]).

Les techniques métaheuristiques sont classées en deux catégories selon le nombre de solutions traités à chaque itération : les algorithmes à base de solution unique et les algorithmes à base de population [14]. La plupart des algorithmes métaheuristiques basés sur la population sont inspirés de phénomènes naturels [9]. Ces approches peuvent être regroupées en quatre catégories principales : celles basées sur l'évolution ( les algorithmes génétiques - GA),

celles basées sur l'essaim ( l'optimisation par essaim particulaire - PSO), celles basées sur la physique ( le recuit simulé - SA) et celles basées sur l'homme (la recherche harmonique - HS). Deux approches contradictoires doivent être équilibrées dans toutes ces techniques pour atteindre des performances convenables : la diversification (exploration de l'espace de recherche) et l'intensification (exploitation de la meilleure solution trouvée) ([15]; [9]).

Les algorithmes métaheuristiques ont leurs propres paramètres spécifiques en plus des paramètres de contrôle communs comme la taille de la population, le nombre de générations et la taille de l'élite [16]. L'efficacité des algorithmes est sensible aux valeurs des paramètres. Un mauvais choix pour les valeurs des paramètres augmentera l'effort de calcul ou conduira à une solution optimale erronée [17]. Cependant, en raison du comportement stochastique de ces techniques, il n'y a aucune garantie de trouver la solution optimale [14]. En outre, ils peuvent souffrir des performances médiocres dans la résolution de certains problèmes. Par ailleurs, le théorème No-Free- Lunch (NFL) confirme qu'il n'existe pas d'algorithme qui puisse être considéré comme le meilleur pour résoudre tous les problèmes d'optimisation [18]. Par conséquent, la réponse à "quel algorithme est le plus approprié pour résoudre le problème" reste ouverte [2]. Ces raisons nous ont motivés à étudier l'efficacité d'algorithmes génétique et l'algorithme symbiotique organism search dans le domaine de l'optimisation temporelle des feux de circulation.

Les approches d'ordonnancement de feux de signalisation souffrent de plusieurs insuffisances dont les plus significatives se déclinent selon les points suivants :

- Elles ne prennent pas en compte la multi-intersection. Cependant, le traitement de ces approches étudient seulement une seule intersection qui ne reflète pas la réalité[67][80][22].
- Le choix des intersections est aléatoire, il serait plus efficace de traiter que les intersections critiques[21].
- Certaines approches traitent une seule intersection d'une ville précisée vue l'utilisation de dataset. Il serait plus judicieux de s'adapter à n'importe quelle ville[55].

Pour pallier à ces insuffisances, nous proposons dans cette thèse deux méthodes d'exploitation optimisées dans la gestion des feux de signalisation dans l'objectif de minimiser le nombre des automobiles en attente.

### OBJECTIF

L'objectif recherché du problème est d'obtenir une programmation optimale globale des feux de circulation qui améliore les conditions de circulation de manière globale. Dans les réseaux urbains, il existe des centaines de carrefours contrôlés par des feux de circulation. Ces feux de signalisation nécessitent un contrôle et une coordination appropriés pour atteindre l'objectif souhaité. Cependant, comment optimiser les horaires de centaines de feux de circulation est devenu un problème complexe et difficile.

Dans ce contexte, nous nous sommes intéressés à la gestion de plusieurs intersections. Notre point de départ sont les études[67][21] présentées dans le cadre de gestion de feux de signalisation. Comme mentionné précédemment, les approches proposées dans ces études présentent des insuffisances dont les plus significatives sont :

- La non prise en compte de traitement de plusieurs intersections ;
- Le choix des intersections à traiter joue un rôle primordiale dans le traitement des attentes des véhicules.

Les problèmes à résoudre et les aspects auxquels nous nous sommes intéressés dans cette thèse sont multiples. Le premier aspect consiste à choisir les intersections critiques par la méthode DIJKSTRA qui est une méthode populaire et qui a fait ces preuves dans d'autres domaines ainsi que l'algorithme de colonie des fourmis(ACO) pour choisir les feux critiques . Le deuxième aspect consiste à utiliser des méta-heuristiques qui sont l'algorithme génétique(GA) et Discret Symbiotic Organism (DSOS) afin d'optimiser les feux de signalisation et en fin de compte minimiser l'attente des voitures.

Les algorithmes génétiques sont appliqués pour résoudre le problème de l'optimisation des feux de signalisation. Les AG sont des techniques d'optimisation inspirées de la théorie de l'évolution et de la génétique. Ils utilisent des opérations génétiques telles que la sélection, la recombinaison et la mutation pour explorer l'espace des solutions et trouver la meilleure configuration des feux de



signalisation. Les AG sont capables de gérer efficacement les problèmes d'optimisation complexes et peuvent fournir des solutions de haute qualité. L'objectif est de trouver la configuration optimale des feux de signalisation qui minimise le nombre des véhicules en attente.

La deuxième approche repose sur l'algorithme de recherche d'organismes symbiotiques discrets (DSOS). Cet algorithme s'inspire des interactions symbiotiques entre différentes espèces dans la nature pour résoudre des problèmes d'optimisation. Ils utilisent des opérations symbiotiques telles que mutualisme, commensalisme et parasitisme pour trouver la meilleure configuration des feux de signalisation.

## ORGANISATION DU MANUSCRIT

Le travail que nous avons mené dans le cadre de la problématique des feux de signalisation est résumé dans le présent document qui est structuré comme suit :

**Chapitre 1 :** Le premier chapitre présente un état de l'art sur les systèmes des feux de signalisation.

**Chapitre 2 :** Le deuxième chapitre présente un état de l'art sur les méta-heuristiques.

**Chapitre 3 :** Le troisième chapitre sera réservé à la description détaillée des stratégies proposées sur l'adaptation des méta-heuristiques sur les systèmes des feux de signalisation.

**Chapitre 4 :** Le quatrième chapitre présentera l'évaluation des stratégies proposées. Les résultats d'expérimentation seront interprétés.

Enfin, Une synthèse et un ensemble de perspectives feront l'objet de notre conclusion.

# Chapitre **1**

## MÉTHODES DE MODÉLISATION ET DE CONTRÔLE DU TRAFIC

### Sommaire

---

<b>1.1</b>	<b>Introduction</b>	19
<b>1.2</b>	<b>Historique</b>	19
<b>1.3</b>	<b>Modélisation et contrôle de trafic</b>	20
1.3.1	Paramètre de transport	20
1.3.2	Modèles de trafic	24
<b>1.4</b>	<b>Système de transport intelligent</b>	27
1.4.1	Communications sans fil	28
1.4.2	Système De Positionnement Global (GPS)	28
1.4.3	Véhicule autonome	29
1.4.4	Communications V2V et V2I	30
1.4.5	Capteur vidéo	31
1.4.6	Capteurs micro-ondes	31
1.4.7	Communications à courte portée	32
1.4.8	Détection de boucle inductive	32
<b>1.5</b>	<b>Contre-mesures pour la congestion du trafic</b>	32
1.5.1	Transport public	32
1.5.2	Planification de l'itinéraire	33
1.5.3	Contrôle des feux de circulation	33
1.5.4	Prédiction du trafic	33

## Méthodes de modélisation et de contrôle du trafic

---

1.5.5	Stratégies de tarification . . . . .	33
1.5.6	Infrastructures routières . . . . .	34
1.6	Conclusion . . . . .	34

---

### 1.1. INTRODUCTION

Le contrôle du trafic est l'un des moyens techniques les plus importants pour réguler le flux du trafic, améliorer la congestion et même réduire les émissions. Son progrès et son développement ont toujours été accompagnés par le développement des technologies de l'informatique et de la science des systèmes. Les progrès récents dans les méthodes de contrôle du trafic ont conduit à des stratégies de contrôle flexibles à utiliser dans un système de contrôle du trafic [19]. L'infrastructure numérisée et informationnalisée du trafic routier métropolitain et la construction des systèmes connexes se sont développées rapidement au cours de la dernière décennie.

Dans ce chapitre, il y a deux parties principales. Dans une première partie, nous nous concentrons sur les méthodes classiques, telles que la description générale du contrôle du trafic et les recherches concernées. Dans la deuxième partie, le système du trafic intelligent (ITS) est présenté, qui est basé sur le développement technologique de la connexion sans fil.

### 1.2. HISTORIQUE

La congestion du trafic est un problème chronique, le premier système de contrôle des feux de circulation a été utilisé à Londres en 1868. Ce système applique manuellement les sémaphores actionnés pour prévenir les accidents en attribuant alternativement la priorité aux véhicules. Un développement important des systèmes de contrôle de la circulation utilisant des feux de signalisation a été réalisé depuis le premier contrôleur de la circulation a été installé, une centaine d'intersections signalisées ("systèmes de contrôle de la circulation de zone"). La première vague verte a été réalisée à Salt Lake City en 1918, et le premier contrôle du trafic régional a été introduit en Toronto en 1960.

L'équipement de contrôle de la circulation a suivi le développement de la technologie. Au tout début, le contrôle de la circulation ont été réalisées par des appareils électromécaniques. Puis, des contrôleurs à base de semi-conducteurs ont été introduits, et de nos jours, des contrôleurs à base de microprocesseur sont utilisés dans le systèmes de contrôle du trafic. Le développement du contrôle régional de la circulation systèmes, surtout depuis 1960, a conduit à l'introduction

d'autres équipements des systèmes de contrôle de la circulation, tels que les ordinateurs, appareils de télécommunication, détecteurs de véhicules, etc. Par conséquent, les stratégies de contrôle du trafic se sont également améliorées depuis l'installation du premier contrôleur de la circulation.

Les chercheurs tentent de mettre à jour les équipements de contrôle du trafic ou d'ajuster l'horaire des feux de signalisation pour améliorer l'efficacité du trafic. En conséquence, les avantages suivants peuvent être obtenus :

- Prouver la grande efficacité du mouvement aux personnes dans la vie de tous les jours.
- Réduire les délais et augmenter le volume de circulation aux intersections.
- Améliorer la sécurité dans la mobilité en diminuant la fréquence et la gravité des accidents.
- L'économie de la consommation de carburant et la protection de l'environnement [19].

### 1.3. MODÉLISATION ET CONTRÔLE DE TRAFIC

#### 1.3.1. PARAMÈTRE DE TRANSPORT

Le système de transport urbain est le moteur des activités économiques dans toutes les communautés urbaines du monde entier et, par conséquent, assure la subsistance des personnes qui y vivent. Les installations de transport urbain typiques comprennent les chemins de fer, les voies navigables, les voies aériennes et les routes. Nous allons présenter les variables à mesurer et analyser le trafic [20].

### HEADWAY

Est une mesure de l'espace temporel entre deux véhicules ( $t_{inter}$ ). Plus précisément, l'écart est le temps qui s'écoule entre l'arrivée du véhicule de tête et le véhicule suivant au point de test désigné. Vous pouvez mesurer l'écart entre deux véhicules en démarrant un chronographe lorsque le pare-chocs avant du premier véhicule franchit le point sélectionné, puis en enregistrant le temps que le pare-chocs avant du deuxième véhicule franchit le point désigné. Comme représenté sur la Figure 1.1 [20].

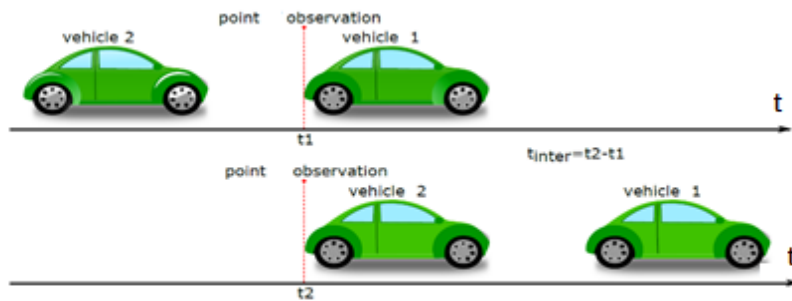


FIGURE 1.1 – Headway.

### DÉBIT DE VÉHICULES (Q)

Le débit est l'un des paramètres du trafic les plus courants. Le débit est la vitesse à laquelle les véhicules passent à un point donné de la chaussée et est normalement exprimé en véhicules par heure. Généralement, on calcule le débit moyen  $Q_m$  pour l'intervalle  $[t_1; t_2]$ . Il peut être mesuré comme le montre l'équation (3.7) et la figure 1.2[20].

$$Q_m(t_1, t_2, X) = \frac{N(t_1, t_2, X)}{t_2 - t_1} \quad (1.1)$$

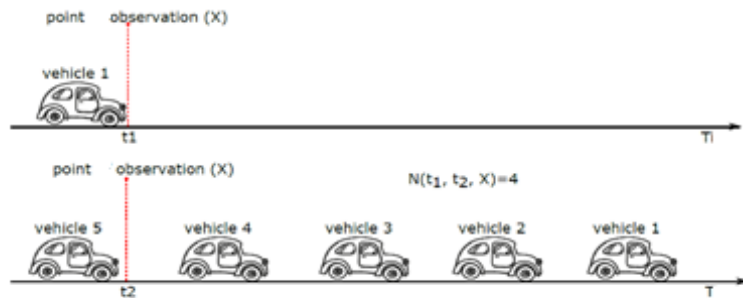


FIGURE 1.2 – Débit de véhicules ( $Q$ ).

### DENSITÉ(D)

La densité fait référence au nombre de véhicules présents sur une longueur de chaussée donnée. Normalement, la densité est rapportée en termes de véhicules par mile ou de véhicules par kilomètre. Des densités élevées indiquent que les véhicules individuels sont très proches les uns des autres, tandis que des densités faibles impliquent de plus grandes distances entre les véhicules. En général,  $D_m$  la densité moyenne est calculée comme indiqué par l'équation (1.2) et la figure 1.3 [20].

$$D_m(X_1, X_2, t) = \frac{N(X_1, X_2, t)}{X_2 - X_1} \quad (1.2)$$

Où le terme  $N(X_1; X_2; t)$  est le nombre de véhicules qui se trouvent dans la section de route entre deux points ( $X_1; X_2$ ) à l'instant  $t$ .

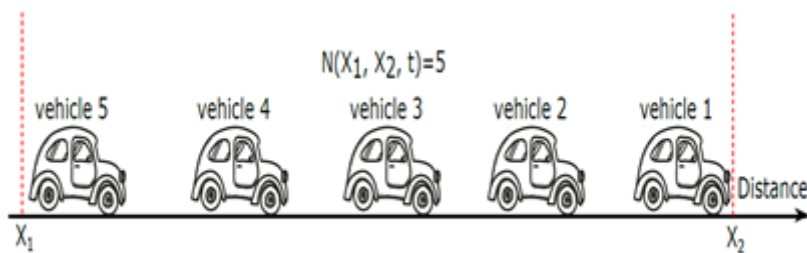


FIGURE 1.3 – Densité.

VITESSE DE VÉHICULE (V)

La vitesse d'un véhicule est définie comme la distance qu'il parcourt par unité de temps. La vitesse moyenne  $V_m$  du véhicule décrit le mouvement moyen du véhicule pendant l'intervalle T.

Elle est mesurée par l'équation (1.3) suivante :

$$V_m = \frac{\int_0^T v(t)dt}{T} \tag{1.3}$$

SCHÉMA FONDAMENTAL DU TRAFIC

Le trafic routier est toujours dans un état spécifique caractérisé par le débit, la densité et la vitesse moyenne. Ce diagramme est illustré dans la Figure 1.4. Lorsque la densité augmente avant le point  $D_c$ , le débit augmente en conséquence. Parce qu'il y a plus de véhicules sur les routes et que tous les véhicules peuvent rouler à grande vitesse. Mais lorsque la densité augmente après le point  $D_c$ , le débit diminue, à cause de la congestion [24].

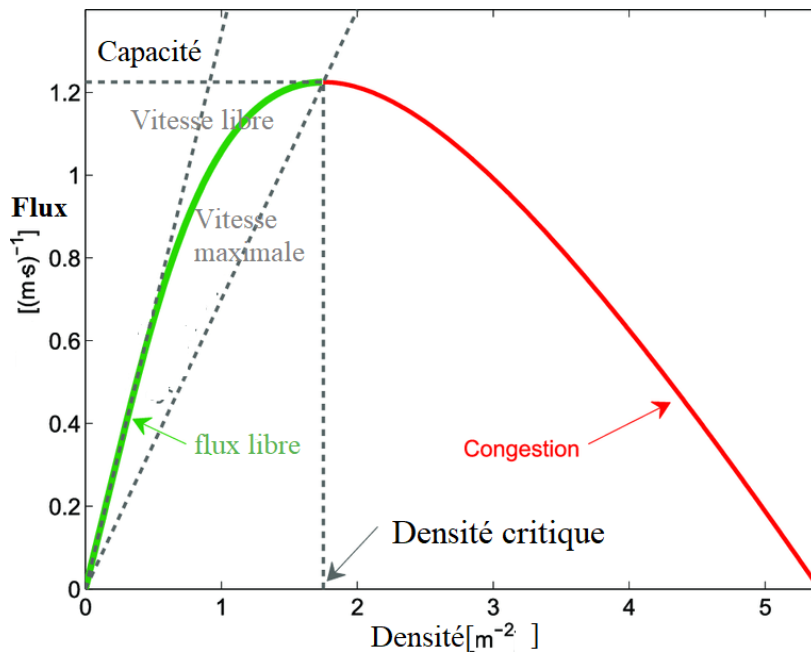


FIGURE 1.4 – Schéma fondamental du trafic.



### 1.3.2. MODÈLES DE TRAFIC

Les modèles de flux du trafic sont des représentations mathématiques du trafic permettant d'estimer les conditions de circulation routière. Les modèles de flux du trafic sont divisés en deux catégories : les modèles microscopiques et les modèles macroscopiques. Des modèles microscopiques représentent chaque véhicule séparément et retracent le comportement de chaque véhicule. Les modèles macroscopiques traitent le trafic comme un fluide continu et étudient la dynamique du trafic de manière agrégée. Les deux modèles sont très actifs dans la recherche et peuvent étudier les phénomènes du trafic [20].

#### MODÈLE DE TRAFIC MICROSCOPIQUE

La modélisation microscopique du trafic routier sont des modèles de suivi de véhicules qui retracent le comportement individuel de chaque véhicule et se concentrent sur la représentation de la relation entre le mouvement de deux véhicules successifs. Les modèles de suivi de voitures sont d'une valeur cruciale pour la simulation du trafic car ils peuvent reproduire le comportement de chaque véhicule individuel avec une grande précision de détails. Par conséquent, le simulateur microscopique est un outil efficace pour valider la conception détaillée des infrastructures de transport et des opérations de transport spécifiques.

Dans les modèles microscopiques, chaque véhicule est indexé avec un numéro en fonction de son ordre d'entrée dans le système simulé. La figure 1.5 illustre un schéma de 4 véhicules dans différents emplacements d'un tronçon de route à un instant  $t$ . Le véhicule  $n$  à l'emplacement  $x_n$  suit le véhicule  $n - 1$  à l'emplacement  $X_n - 1$ , et le véhicule  $X_n + 1$  suit le véhicule  $n$  à l'emplacement  $X_n$  [20].

La modélisation microscopique du trafic routier sont des modèles de suivi de véhicules qui retracent le comportement individuel de chaque véhicule et se concentrent sur la représentation de la relation entre le mouvement de deux véhicules successifs [20].

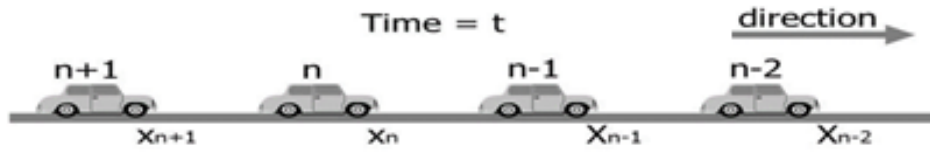


FIGURE 1.5 – Indice d'ordre des véhicules dans les modèles microscopiques..

L'emplacement, la vitesse et l'accélération sont des variables considérées dans les modèles microscopiques. En prenant le véhicule  $n$  dans la Figure 6 comme exemple, et en supposant qu'il se déplace sur une distance de  $\Delta x$  de l'instant  $t$  à l'instant  $t + \Delta t$ , sa vitesse  $v_n$  et son accélération  $a_n$  à l'instant  $t$  peut être calculé par les équations suivantes :

$$V_n = \frac{dx_n}{dt} = \lim_{\Delta t \rightarrow 0} \frac{\Delta x}{\Delta t} \quad (1.4)$$

$$a_n = \frac{dv_n}{dt} = \frac{d^2x_n}{dt^2} \quad (1.5)$$

Le mécanisme des modèles de suivi de véhicule est essentiellement un processus stimulus-réponse qui peut être généralement exprimé comme [20] :

$$Réponse(plustard) = Sensitivité \times Stimulus(maintenant)$$

Lorsqu'il reçoit un stimulus de la circulation, un conducteur peut répondre par une série d'actions telles que le freinage, l'accélération et le changement de voie. Les modèles de suivi de voiture peuvent être différenciés par différents éléments de stimulation tels que la distance de suivi de sécurité, la vitesse relative et l'espacement relatif, entre autres.

Les modèles de flux du trafic microscopiques nécessitent, en plus des paramètres requis par des modèles macroscopiques, des informations finement détaillées sur les paramètres de performance du véhicule et du comportement du conducteur, ainsi que sur la géométrie et le tracé de la route [20].

En présence d'applications de systèmes de transport intelligents (ITS) il a été remarqué que les modèles de simulation microscopiques sont adaptés modéliser à la fois les interactions des véhicules et les réactions des conducteurs lorsqu'ils sont exposés à de telles applications. La possibilité de modéliser le choix d'itinéraire dans des modèles de microsimulation est importante lors de l'évaluation d'une application ITS qui aide les conducteurs à décider de leurs itinéraires via des messages en route [20].

### MODÈLE DE TRAFIC MACROSCOPIQUE

La modélisation macroscopique du trafic suppose un nombre suffisamment important de véhicules sur une route pour que chaque flux de véhicules puisse être traité comme circulant dans un tube ou un flux. Trois variables sont importantes dans la modélisation macroscopique du trafic [25].

Ces variables sont :

- Le débit,  $q(x, t)$  : le nombre de véhicules passant par un point fixe  $x$  par unité de temps.
- La vitesse de circulation,  $v(x, t)$  : la distance parcourue par unité de temps.
- La densité du trafic,  $\rho(x, t)$  : le nombre de véhicules dans une ligne du trafic de longueur donnée [25].

Ces variables du trafic sont connectées de la manière suivante :

$$q(x, t) = \rho(x, t)v(x, t) \quad (1.6)$$

Il existe également une autre relation entre ces variables. Cette relation est basée sur la conservation des véhicules :

$$\frac{\delta \rho}{\delta t} + \frac{\delta(\rho v)}{\delta x} = 0 \quad (1.7)$$

Les équations constituent un modèle indépendants et trois variables inconnues. Les modèles du trafic macroscopiques utilisent différentes équations de vitesse pour compléter la description de flux du trafic [25].

### 1.4. SYSTÈME DE TRANSPORT INTELLIGENT

La gestion du trafic est devenue l'une des préoccupations majeures de toute ville urbaine avec l'augmentation du nombre de véhicules. Une solution peut être une augmentation de la croissance des infrastructures, mais elle est inefficace en termes de coût, d'effort et de temps. Par conséquent, les pays du monde entier envisagent d'utiliser les technologies TIC pour développer des systèmes efficaces de gestion du trafic [26].

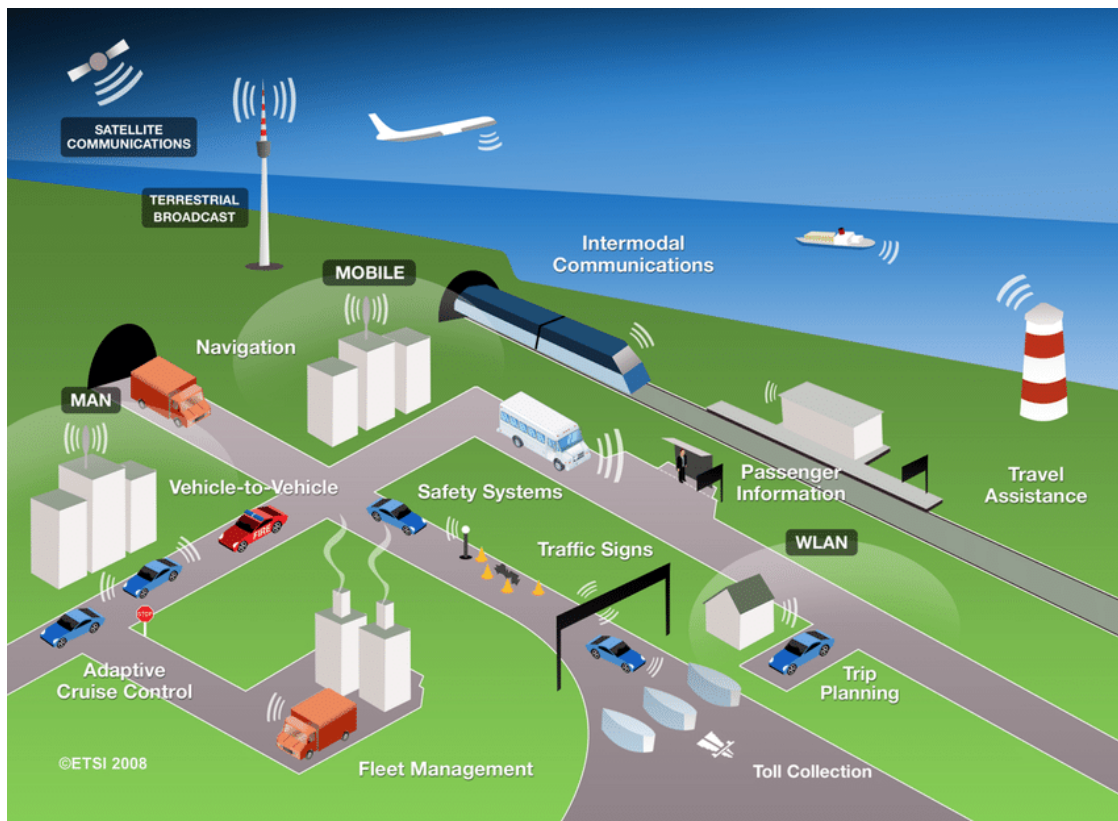


FIGURE 1.6 – Système de transport intelligent.

Le système de transport intelligent intègre les technologies actuelles et en croissance afin d'améliorer les conditions de transport, la sécurité et les services. Nous aborderons ici les technologies les plus utilisées dans ce secteur.

### 1.4.1. COMMUNICATIONS SANS FIL

Les réseaux sans fil utilisent des ondes radio ou des micro-ondes pour transférer et partager les données entre les nœuds après les avoir connectés. Les progrès de la communication sans fil ont permis aux réseaux de capteurs sans fil (WSN) d'avoir un large éventail d'applications potentielles, notamment le défi national, la surveillance de l'environnement, les systèmes médicaux, le suivi des cibles, la détection de l'habitat et la détection des incendies [26]. Les avantages significatifs du WSN dans les système de transport intelligent sont indiqués ci-dessous [26] :

- À l'aide des capteurs, le conducteur peut échanger des informations en temps réel et peut réagir à l'avance aux situations d'urgence.
- La structure du nœud capteur est assez simple et ne coûte pas cher, ce qui le rend pratique pour un grand nombre de déploiements.
- Les nœuds de capteurs peuvent surveiller le trafic en temps réel et envoyer les informations au centre, ce qui accélère le processus de prise de décision.
- En collaborant avec le multi-capteur, il est possible d'améliorer la précision et la fiabilité de la détection.

### 1.4.2. SYSTÈME DE POSITIONNEMENT GLOBAL (GPS)

Le GPS, connu à l'origine sous le nom de NAVSTAR GPS, est un système de radio navigation par satellite conçu et développé par le département américain de la défense en tant qu'aide à la navigation.

La planification de ce système a commencé dans les années 1960 et a atteint sa pleine capacité opérationnelle en 1995. Ce système permet à un nombre illimité de récepteurs GPS situés n'importe où sur la surface de la terre et en vue des satellites GPS de déterminer avec précision la position, la vitesse et le temps.

L'architecture du système GPS se compose de trois segments :

1. Le segment espace avec 24 satellites assurant une couverture mondiale,
2. Le segment de contrôle opérationnel - qui surveille et contrôle le segment spatial,
3. Le segment utilisateur [27].

### 1.4.3. VÉHICULE AUTONOME

Les véhicules autonomes s'appuieront sur des capteurs (par exemple, un radar) pour obtenir des informations, telles que la distance par rapport au véhicule intermédiaire précédent. Avec l'aide de la technologie connectée, les véhicules autonomes acquerront plus d'informations (par exemple, les vitesses et les positions des véhicules plus en avant) et seront capables de conduire un processus de conduite plus rapide, plus fluide et plus fiable [28].



FIGURE 1.7 – Véhicule autonome.

#### 1.4.4. COMMUNICATIONS V2V ET V2I

La communication futuriste vise à permettre aux véhicules, aux infrastructures de transport en bordure de route, aux téléphones intelligents pour piétons et à d'autres appareils de la communication. Les ITS offrent la possibilité d'échanger la vitesse, l'angle de cap, la position et d'autres conditions liées à l'environnement entre véhicules et avec les infrastructures intelligentes environnantes. Dans cette configuration intelligente, les véhicules et les utilisateurs communiquent et échangent des données avec des entités aléatoires non fiables (comme des véhicules, des feux de circulation intelligents ou des piétons) qu'ils ne connaissent pas ou n'ont pas rencontrés auparavant.

Véhicule à véhicule (V2V) et véhicule à infrastructure (V2I) communication sont deux propositions d'innovations technologiques qui peut changer le transport actuel. Le V2V permettra aux véhicules d'échanger des informations sur la vitesse, l'emplacement, la position, la direction, ou l'état des freins avec d'autres véhicules environnants où la réception les véhicules regrouperont ces messages et prendront des décisions intelligentes. Ces applications embarquées préviendront les conducteurs des accidents, vitesse excessive, circulation lente devant, conducteur agressif, angle mort ou danger routier. V2I permettra aux unités routières ou au trafic infrastructures pour transmettre des informations sur la hauteur autorisée du pont, la fusion du trafic, l'avertissement de zone de travaux ou la signalisation des risques routiers [29].

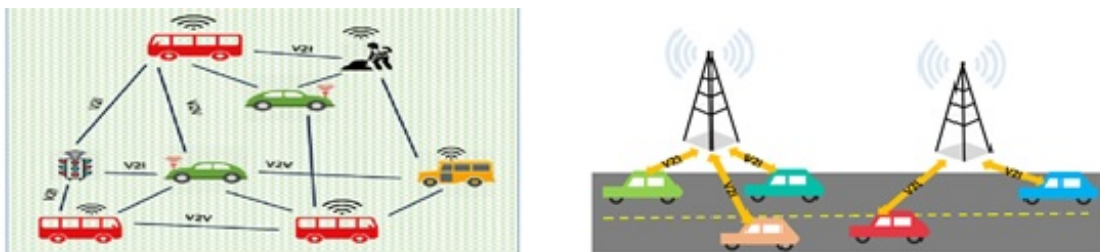


FIGURE 1.8 – Communication V2V et Communication V2I

### 1.4.5. CAPTEUR VIDÉO

Il existe deux processus pour les mesures de trafic sont développés à partir de la vidéo, Le premier processus permet de détecter les plaques d'immatriculation des automobiles, il utilise des algorithmes de reconnaissance de caractère et basé sur les matériels suivants (caméra à champ proche, caméra haute définition, la plaque d'immatriculation d'un véhicule).

Le deuxième processus permet de détecter les automobiles avec l'analyse des images vidéo (suivre le mouvement des pixels sur plusieurs trames consécutives, analyse colorimétrique, réduction du bruit ex. ombres de véhicules [30]).

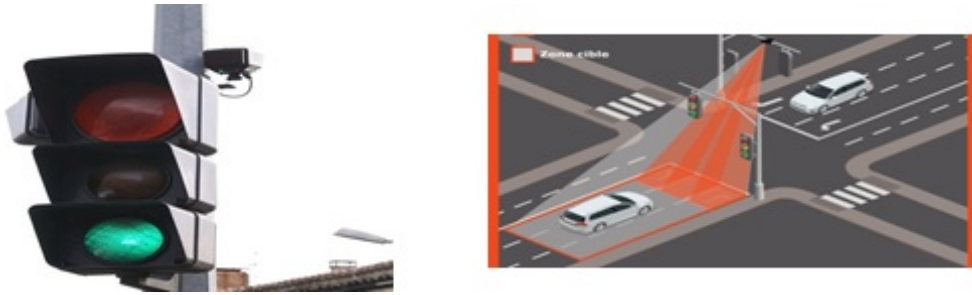


FIGURE 1.9 – Capteur vidéo

### 1.4.6. CAPTEURS MICRO-ONDES

Les capteurs micro-ondes sont appelés RADAR ( Radio Detection And Ranging). Ils ont utilisé des antennes directionnelles pour émettre des ondes électromagnétiques. Il existe deux types de capteur pour les équipements routiers le radar à ondes continues (radar doppler) et le radar à ondes continues modulées en fréquence. Le radar à ondes continues calcule la différence de fréquences appelée « fréquence doppler » entre l'onde émise et l'onde réfléchi. Le radar à ondes continue modulées en fréquence utilise la modulation en fréquence. La différence de temps entre l'émission et la réception du signal détermine la distance entre le radar et le véhicule, en mouvement ou à l'arrêt [31].



### 1.4.7. COMMUNICATIONS À COURTE PORTÉE

La communication dédiée à courte portée a alloué 75 MHz de spectre à 5,9 GHz qui est entièrement utilisé pour le véhicule à véhicule et l'infrastructure à communication des véhicules afin d'améliorer la fluidité du trafic [32].

### 1.4.8. DÉTECTION DE BOUCLE INDUCTIVE

Un détecteur de véhicule à boucle inductive induire un courant électrique dans un fil afin de détecter les véhicules. Ce système est assez rentable et est systématiquement utilisé dans les applications de circulation et de stationnement d'aujourd'hui.

Le principe de fonctionnement de la détection de boucle inductive est qu'il y a un ou plusieurs tours de fil isolé qui sont placés dans une découpe peu profonde dans la chaussée. Le système se compose d'une unité électronique dans une armoire de commande. Du fait du passage d'un véhicule sur la boucle, l'induction du fil est modifiée ce qui provoque le changement de fréquence. Le changement de fréquence fait que l'unité électronique envoie un signal au contrôleur qui indique la présence du véhicule.

La détection de boucle inductive est utilisée pour connaître la présence, le passage, l'occupation des véhicules etc. On peut aussi connaître le nombre de véhicules traversant une zone particulière en utilisant ce système [32].

## 1.5. CONTRE-MESURES POUR LA CONGESTION DU TRAFIC

Voici quelques stratégies pour améliorer les embouteillages :

### 1.5.1. TRANSPORT PUBLIC

Il s'agit d'un service de transport partagé de passagers qui peut réduire considérablement le nombre de véhicules privés. Les transports en commun comprennent le bus, le train, le train à grande vitesse (HSR), le tram, etc [33].

### 1.5.2. PLANIFICATION DE L'ITINÉRAIRE

Cette méthode permet de planifier l'itinéraire avant de voyager à l'avance. Il peut utiliser les données relatives au trafic pour calculer le temps de trajet prévu, et également éviter les tronçons de route accidentés. Il peut alors choisir l'un des meilleurs itinéraires possibles et éviter les embouteillages. Il peut également être associé au système de positionnement global (GPS) pour ajuster dynamiquement le trajet pendant le voyage [33].

### 1.5.3. CONTRÔLE DES FEUX DE CIRCULATION

L'ajustement de la période des feux de circulation est une méthode pour réduire le nombre de voitures en file d'attente. De nos jours, la plupart des systèmes de contrôle des feux de circulation utilisent des paramètres de période différents entre les heures de pointe et les heures creuses [33].

### 1.5.4. PRÉDICTION DU TRAFIC

Les techniques de prévision du trafic utilisent des données du trafic historiques obtenues à partir de capteurs ou de caméras. Les états du trafic futurs prédits sont ensuite fournis au système de planification d'itinéraire ou de réglage des feux de circulation [33].

### 1.5.5. STRATÉGIES DE TARIFICATION

En facturant des redevances aux usagers de la route à certains moments, cela peut limiter le volume de véhicules accédant aux routes. Par exemple, il est maintenant pratiqué à Singapour, à Londres et dans certaines autres villes. Un autre système s'appelle « Cap and Trade », qui n'autorise que les voitures immatriculées qui ont payé de l'argent sur la route. Faire payer des frais de stationnement plus élevés dans les sections congestionnées est également une bonne stratégie pour réduire le désir de l'utilisateur de traverser ou d'entrer dans une section populaire facilement encombrée [33].

### 1.5.6. INFRASTRUCTURES ROUTIÈRES

Le gouvernement ou une organisation apparentée peut élargir ou construire de nouvelles routes pour augmenter le nombre de routes, ou repenser les routes pour réduire les jonctions, telles que les routes surélevées ou souterraines. Il peut également construire des autoroutes sans feux de circulation pour améliorer la situation du trafic [33].

### 1.6. CONCLUSION

Les systèmes de contrôle des feux de circulation pour les intersections signalisées sont l'un des composants les plus importants des systèmes de contrôle du trafic urbain. Son efficacité affecte directement la sécurité des participants et le fonctionnement efficace du système de contrôle de la circulation. Par conséquent, ce chapitre a fourni un aperçu général des systèmes de contrôle des feux de circulation, des méthodes de modélisation et de contrôle du trafic, des paramètres de contrôle du trafic, des techniques d'intelligence informatique utilisées pour optimiser les performances des systèmes de contrôle des feux de circulation, et contre-mesures pour la congestion du trafic.

# Chapitre 2

## MÉTHODES BIO-INSPIRÉES

### Sommaire

---

2.1	Introduction . . . . .	37
2.2	Techniques d'optimisation des métaheuristiques . . . . .	38
2.3	Algorithme d'optimisation des colonies des fourmis (ACO) . . . . .	40
2.3.1	Principe . . . . .	40
2.3.2	Fonctionnement d'Algorithme . . . . .	41
2.3.3	Travaux connexes . . . . .	42
2.4	Algorithme d'optimisation par essais de particules (PSO) . . . . .	43
2.4.1	Principe . . . . .	43
2.4.2	Fonctionnement d'Algorithme . . . . .	44
2.4.3	Travaux connexes . . . . .	45
2.5	Colonie d'abeilles artificielles (ABC) . . . . .	46
2.5.1	Principe . . . . .	46
2.5.2	Fonctionnement d'algorithme . . . . .	46
2.5.3	Travaux connexes . . . . .	46
2.6	Recherche de coucou (CS) . . . . .	47
2.6.1	Principe . . . . .	47
2.6.2	Fonctionnement . . . . .	48
2.6.3	Travaux connexes . . . . .	49
2.7	Algorithme de chauve-souris (Bat algorithm (BA)) . . . . .	50
2.7.1	Principe . . . . .	50
2.7.2	Fonctionnement . . . . .	50

2.7.3	Travaux connexes . . . . .	51
<b>2.8</b>	<b>Système immunitaire artificiel (AIS) . . . . .</b>	<b>51</b>
2.8.1	Principe . . . . .	51
2.8.2	Fonctionnement . . . . .	52
2.8.3	Travaux connexes . . . . .	52
<b>2.9</b>	<b>Optimiseur des loups gris (GWO) . . . . .</b>	<b>53</b>
2.9.1	Principe . . . . .	53
2.9.2	Fonctionnement . . . . .	54
2.9.3	Travaux connexes . . . . .	55
<b>2.10</b>	<b>Recherche d'harmonie (HS) . . . . .</b>	<b>55</b>
2.10.1	Principe . . . . .	55
2.10.2	Fonctionnement . . . . .	56
2.10.3	Travaux connexes . . . . .	56
<b>2.11</b>	<b>Conclusion . . . . .</b>	<b>56</b>

---

### 2.1. INTRODUCTION

Le contrôle des feux de circulation est essentiel à la gestion du trafic urbain car ses performances affectent directement l'efficacité du système de circulation. Récemment, des approches d'optimisation ont été utilisées dans des modèles de contrôle du trafic pour augmenter les performances des systèmes de contrôle des feux de circulation. L'objectif principal d'une optimisation des feux de circulation est d'améliorer considérablement les performances du carrefour en minimisant le retard, la longueur de la file d'attente, le nombre d'arrêts, les émissions et en maximisant le flux de trafic et la vitesse moyenne sur le réseau.

Le réglage de la synchronisation des feux de circulation dans un réseau routier contrôlé par la signalisation implique la détermination du temps de cycle, des fractionnements de temps vert et des décalages. L'optimisation des feux de circulation peut optimiser une partie ou la totalité de ces valeurs en fonction des paramètres de trafic observés, tels que le débit et la longueur de la file d'attente. Un ou plusieurs objectifs peuvent être impliqués dans les modèles d'optimisation des feux de circulation.

L'optimisation est le processus de recherche des meilleures solutions qui maximise ou minimise d'une fonction [34]. Les méthodes de recherche optimales sont connues sous le nom des méthodes de programmation mathématique. Dans chaque problème d'optimisation, il y a les éléments suivants : 1) l'espace de recherche qui est l'ensemble des solutions possibles. 2) la fonction de coût (fonction objectif) qui est le modèle utilisé pour évaluer les solutions. 3) les contraintes (éventuellement vides) qui sont un ensemble de conditions pour les variables d'entrée qui doivent être satisfaites. [35]

Une variété de problèmes du monde réel peuvent être formulés comme un problème d'optimisation. En effet, les techniques d'optimisation sont largement utilisées pour résoudre de nombreux problèmes du monde réel dans plusieurs domaines, tels que les systèmes de contrôle automatique, la conception électronique, les problèmes de conception chimique, mécanique et civile [36].

Le choix de la technique dépend de la nature et des caractéristiques du problème à résoudre [9].

Les méthodes d'optimisation peuvent être classées de plusieurs manières, l'une de ces classifications les divise en méthodes exactes et méthodes heuristiques en fonction de la complexité du problème [10]. Les méthodes exactes, telles que la programmation dynamique, la programmation par contraintes, les méthodes de backtracking, les méthodes branch-and-X(branch-and-bound, branch-and-cut, branch-and-price) garantissent de trouver la solution optimale au problème à résoudre, ils sont adaptés pour résoudre de petites instances de problèmes difficiles où le temps requis augmente de manière polynomiale par rapport aux dimensions du problème [37]. Alors que les méthodes heuristiques ne garantissent pas qu'une solution globalement optimale puisse être trouvée dans une classe de problèmes, elles peuvent trouver une solution "quasi optimale" dans un délai raisonnable [9]. Dans les problèmes d'optimisation combinatoire avec un espace de recherche de grande dimension, trouver toutes les solutions possibles consomme du temps et des ressources. En recherchant sur un large ensemble de solutions réalisables, les méthodes heuristiques peuvent souvent trouver de bonnes solutions avec moins d'effort de calcul et sont donc appropriées pour résoudre cette classe de problèmes [10].

En général, les méthodes heuristiques peuvent être classées en deux types : Les heuristiques spécifiques et les métaheuristiques. Les méthodes heuristiques spécifiques sont dépendant du problème et sont développées pour résoudre des problèmes très spécifiques. D'un autre côté, les méthodes métaheuristiques sont indépendantes du problème et conviennent donc à la résolution d'un large éventail de problèmes [9].

### **2.2. TECHNIQUES D'OPTIMISATION DES MÉTAHEURISTIQUES**

Les techniques métaheuristiques sont un type de méthodes d'optimisation stochastique où un certain degré d'aléatoire et de probabilité est utilisé pour trouver les solutions (quasi) optimales[35]. Ces méthodes explorent l'espace de recherche pour trouver de bonnes solutions sans garantir la solution optimale [14].

Dans de tels problèmes, nous n'avons pas d'informations préalables sur la façon dont la meilleure solution semble. Lorsqu'on donne une solution candidate, sa qualité ou son adéquation peut être évaluée à l'aide de la fonction objective. [14]

Les algorithmes métaheuristiques peuvent être classés de plusieurs façons. L'une des catégorisations les plus populaires est celle qui dépend du nombre de solutions traitées à chaque itération. Les algorithmes basés sur une solution unique (S) sont des algorithmes qui manipulent une solution à chaque itération du processus d'optimisation, tandis que les algorithmes basés sur la population (P) manipulent un ensemble de solutions (appelé population) à chaque itération du processus d'optimisation [14]. Le recuit simulé (SA), la recherche Tabou (TS) et le grand déluge (GD) sont des exemples d'algorithmes métaheuristiques basés sur la solution unique (S). L'algorithme génétique (GA), la colonie d'abeilles artificielle (ABC), l'optimisation par colonies de fourmis (ACO), l'optimisation par essaims de particules (PSO) sont des exemples d'algorithmes métaheuristiques basés sur la population (P). De plus, la plupart des algorithmes métaheuristiques basés sur la population sont inspirés par la nature [9], on peut les classer en groupes (voir figure 2.1) : les algorithmes basés sur l'évolution (par ex. GA, ES), les algorithmes basés sur l'essaimage (par exemple, PSO, TLBO, Jaya, et ACO), les algorithmes basés sur la physique (par exemple, SA, TLBO, Jaya, et ACO) (par exemple, SA, GSA) et humaine (par exemple, HS) [38].

Dans les algorithmes métaheuristiques basés sur la population (P), le processus d'optimisation est accompli en deux phases principales : exploration (ou diversification) et exploitation (ou intensification). Dans l'exploration, un grand échelle de régions de l'espace de recherche est examinée pour générer diverses solutions, de manière à réduire le risque de piège dans un minimum local [10]. D'autre part, exploiter signifie examiner plus attentivement les régions prometteuses pour trouver de meilleures solutions [9]. Cependant, un bon compromis entre ces deux composantes est nécessaire pour atteindre l'optimalité globale [38].

Les algorithmes métaheuristiques sont des algorithmes probabilistes et nécessitent donc leurs propres paramètres spécifiques en plus des paramètres de contrôle communs [39]. Ces algorithmes sont très sensibles aux réglages des paramètres. L'absence de réglage fin des valeurs de ces paramètres affectera négativement les performances de l'algorithme employé [40]. Considérant ce fait, des algorithmes sans paramètres récemment publiés, appelés TLBO et Jaya, ont été introduits et ont montré une bonne performance dans la résolution d'une variété de problèmes [41], [42].



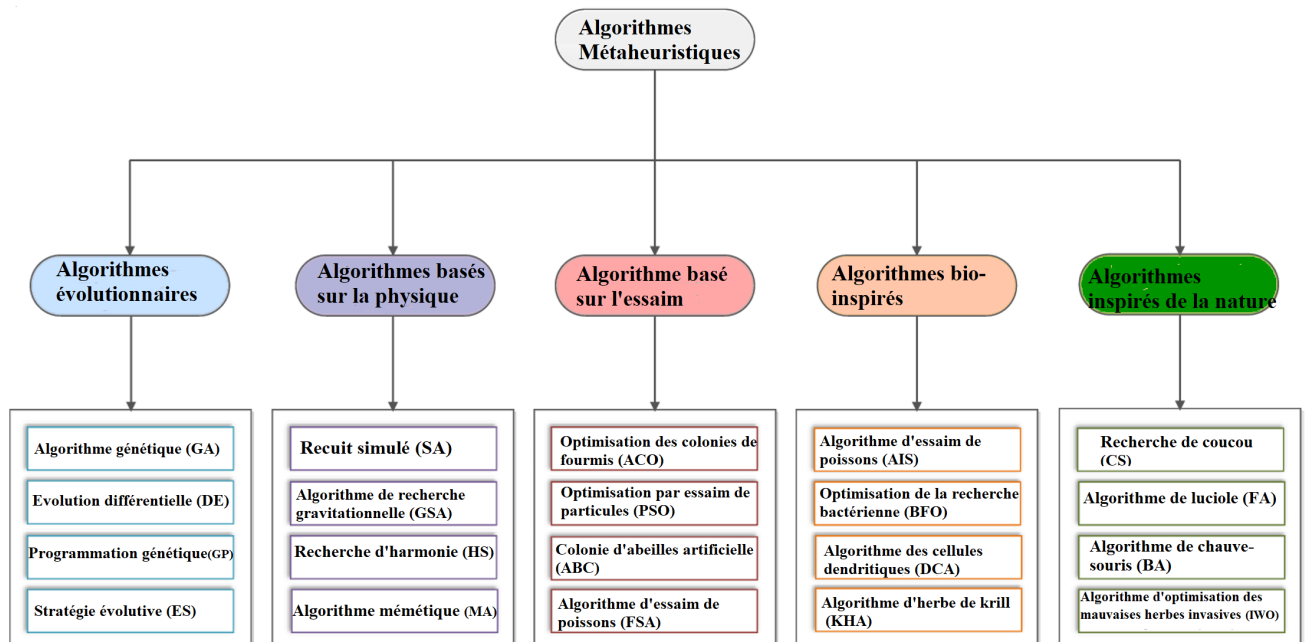


FIGURE 2.1 – Classification des algorithmes métaheuristiques.

## 2.3. ALGORITHME D'OPTIMISATION DES COLONIES DES FOURMIS (ACO)

### 2.3.1. PRINCIPE

Les colonies de fourmis, et plus généralement les sociétés sociales d'insectes, sont des systèmes distribués malgré la simplicité de leurs individus, présentent une organisation sociale très structurée. Grâce à cette organisation, les colonies de fourmis peuvent accomplir des tâches complexes qui est dans certains cas, dépassent de loin les capacités individuelles d'une seule fourmi. [45]

Le domaine des « algorithmes de fourmis » étudie des modèles dérivés de l'observation du comportement réel de fourmis et utilise ces modèles comme source d'inspiration pour la conception de nouveaux algorithmes pour la résolution de problèmes d'optimisation et de contrôle distribué.

L'idée principale est que les principes d'auto-organisation qui permettent le comportement hautement coordonné des fourmis réelles peuvent être exploités pour coordonner des populations d'agents artificiels qui collaborent pour résoudre des problèmes de calcul. [45]

Plusieurs aspects différents du comportement des colonies de fourmis ont inspiré différents types d'algorithmes de fourmis. Les exemples sont la recherche de nourriture, la division du travail, le tri du couvain et le transport coopératif. Dans tous ces exemples, les fourmis coordonnent leurs activités via la stigmergie, une forme de communication indirecte médiée par des modifications de l'environnement. L'idée derrière les algorithmes de fourmis est alors d'utiliser une forme de stigmergie artificielle pour coordonner des sociétés d'agents artificiels. L'un des exemples les plus réussis d'algorithmes de fourmis est connu sous le nom « optimisation des colonies de fourmis (ACO) ». [45]

### 2.3.2. FONCTIONNEMENT D'ALGORITHME

- Une fourmi parcourt plus ou moins au hasard l'environnement autour de la colonie ;
- Si celle-ci découvre une source de nourriture, elle rentre plus ou moins directement au nid, en laissant sur son chemin une piste de phéromones ;
- Ces phéromones étant attractives, les fourmis passant à proximité vont avoir tendance à suivre, de façon plus ou moins directe ;
- En revenant au nid, ces mêmes fourmis vont renforcer la piste ;
- Si deux pistes sont possibles pour atteindre la même source de nourriture, celle étant la plus courte sera, dans le même temps, parcourue par plus de fourmis que la longue piste ;
- La piste courte sera donc de plus en plus renforcée, et donc de plus en plus attractive ;
- La longue piste, elle, finira par disparaître, les phéromones étant volatiles ;
- L'ensemble des fourmis a donc déterminé et « choisi » la piste la plus courte [44].

### 2.3.3. TRAVAUX CONNEXES

L'algorithme d'optimisation des colonies de fourmis (ACO) s'inspire du comportement des fourmis dans la recherche de ressources alimentaires. La théorie copie la façon dont les fourmis laissent des empreintes de phéromones pour que les autres fourmis les suivent lorsqu'elles sont à la recherche de ressources alimentaires [46].

Ce faisant, les fourmis parviennent à trouver le chemin le plus court (ou optimal) lors des allers-retours entre la colonie et la source de nourriture. De manière analogue à cela, l'ACO suggère que les agents (c'est-à-dire les fourmis) sont en communication directe tout en cherchant des informations à l'aide de sentiers artificiels (les phéromones). Les pistes sont construites sur la base d'une expérience de recherche probabiliste qui peuvent rapidement distribuer les résultats à d'autres agents grâce à des processus de rétroaction [48].

L'algorithme a été proposé pour la première fois par Dorigo et al. dans [49], puis élargi pour accueillir des implémentations dans divers autres domaines [43], tels que le problème du voyageur de commerce [50], le routage de véhicules [43], l'informatique routage de réseaux [46], coloration de graphes [51] etc. Dans les applications relatives aux systèmes de transport; l'algorithme de colonie de fourmis vise à trouver le chemin le plus court ou le plus optimisé entre une source et une destination.

TABLE 2.1 – Travaux Connexes sur les Algorithmes de gestion de trafic basé sur ACO.

Ref	Titre de papier	Algorithme utilisé	Paramètres à optimiser	Nombre d'intersection	Implémentation
[60]	Ant colony algorithm for traffic signal timing optimization	Ant Colony Optimization	Time Delay	Une seule intersection	Simulateur implémenté
[61]	Identification and optimization of traffic bottleneck with signal timing	Ant Colony Optimization	réduire le risque de variance élevée de l'indicateur de goulot d'étranglement	1 intersection	Simulateur implémenté

## 2.4. ALGORITHME D'OPTIMISATION PAR ESSAIMS DE PARTICULES (PSO)

### 2.4.1. PRINCIPE

L'optimisation des essaims de particules [10] s'inspire du comportement de groupe collectif d'organismes décentralisés et auto-organisés (systèmes ou agents) voir la figure 2.2. Ces organismes interagissent les uns avec les autres pour atteindre un objectif commun (souvent optimal) décidé sur la base des commentaires des membres individuels du groupe. Les groupes d'organismes comme les colonies de fourmis, les troupes d'oiseaux, les troupes d'animaux, la croissance bactérienne et les bancs de poissons ont tous un comportement d'essaim. Pour imiter ce comportement, l'algorithme a été présenté pour la première fois comme une technique d'optimisation stochastique basée sur la population par Eberhart et Kennedy en 1995 [52].

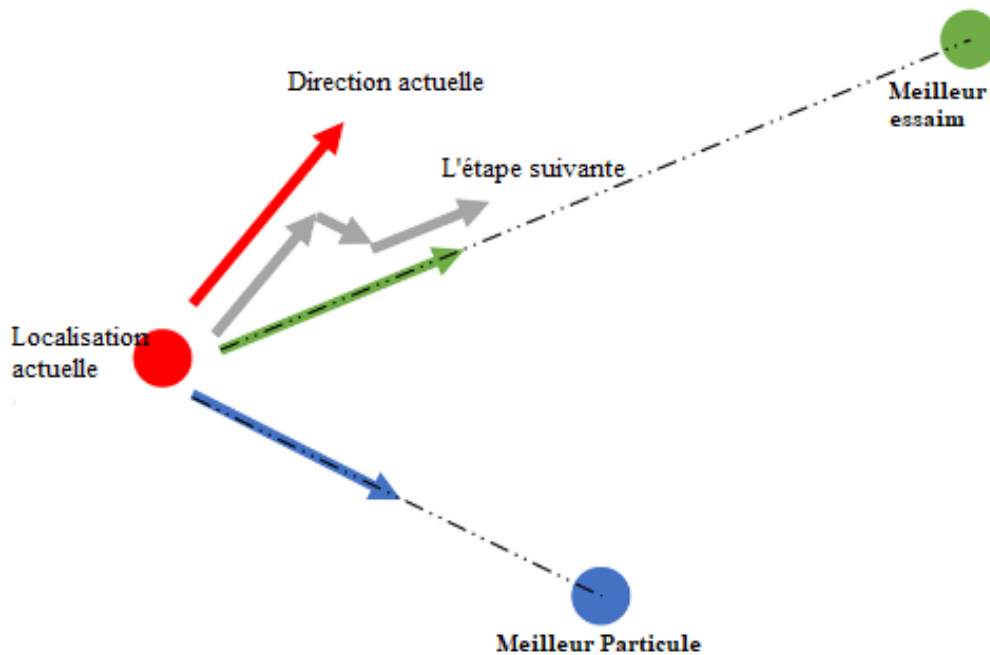


FIGURE 2.2 – Illustration d'optimisation d'essaim de particules.

### 2.4.2. FONCTIONNEMENT D'ALGORITHME

- La vitesse actuelle  $V_k$ ;
- La meilleure solution  $P_i$ ;
- La meilleure solution obtenue dans son voisinage  $P_g$ .

Cela donne l'équation de mouvement suivante :

$$V_{k+1} = wV_k + b_1(P_i - X_K) + b_2(P_g - X_K) \quad (2.1)$$

$$X_{K+1} = X_K + V_{K+1} \quad (2.2)$$

Avec :

- $X_k$  sa position actuelle
- $W$  inertie
- $b_1$  tiré aléatoirement dans  $[0,1]$
- $b_2$  tiré aléatoirement dans  $[0,1]$ .

Chaque particule est déplacée par deux forces élastiques, l'une l'attirant avec une amplitude aléatoire vers l'emplacement le plus approprié rencontré jusqu'à présent par la particule, et l'autre l'attirant avec une amplitude aléatoire vers le meilleur emplacement rencontré par n'importe quel des voisins sociaux de la particule dans l'essaim. Si le problème est à  $N$  dimensions, la position et la vitesse de chaque particule peuvent être représentées comme un vecteur à  $N$  composantes (une pour chaque dimension). En partant du vecteur vitesse,  $v = (v_1, \dots, v_N)$ , chaque composante,  $v_i$ , est donnée par :

$$v_i(t+1) = wv_i(t) + Z_1R_1(x_{si} - x_i(t)) + Z_2R_2(x_{pi} - x_i(t)) \quad (2.3)$$

Où  $x_{si}$  est la  $i$ ème composante du meilleur point visité par les voisins de la particule,  $x_i(t)$  est la  $i$ ème composante de l'emplacement actuel de la particule,  $x_{pi}$  est la  $i$ ème composante de son record personnel,  $R_1$  et  $R_2$  sont deux variables aléatoires indépendantes uniformément réparties dans  $[0, 1]$ , est une constante appelée poids d'inertie,  $Z_1$  et  $Z_2$  sont deux constantes, appelées coefficients d'accélération, qui contrôlent la proportion relative de cognition et d'interaction sociale dans l'essaim [53].

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2.4)$$

Finalement, l'essaim dans son ensemble, comme une volée d'oiseaux cherchant collectivement de la nourriture, est susceptible de se déplacer à proximité du meilleur emplacement. L'équation alternative suivante de mise à jour de la vitesse a été développée et est également largement utilisée :

$$v_i(t + 1) = k(v_i(t) + Z_1 R_1(x_{si} - x_i(t)) + Z_2 R_2(x_{pi} - x_i(t))) \quad (2.5)$$

Où  $k$  est une constante appelée coefficient de constriction. Si  $Z_1$ ,  $Z_2$  et  $k$  (ou  $w$ ) sont correctement choisis, le PSO est garanti stable sans nécessiter de contraintes particulières (par exemple, limitation des vitesses et des positions)[53].

### 2.4.3. TRAVAUX CONNEXES

En 1989 Beni Wang [54] ont développé Swarm Intelligence dans le contexte des systèmes robotiques cellulaires . Les membres individuels de l'essaim sont à la recherche d'une solution potentielle à n'importe quel point dans un espace multidimensionnel du paramètre (optimisation à valeur réelle), et donc la pertinence de la solution candidate basée sur une fonction de fitness. Lorsque nécessaire, les agents tentent de déterminer des meilleures solutions qui guident les agents vers des solutions plus optimales.

TABLE 2.2 – Travaux Connexes sur les Algorithmes de gestion de trafic basé sur PSO.

Ref	Titre de papier	Algorithme utilisé	Paramètres à optimiser	Nombre d'intersection	Implémentation
[55]	Swarm intelligence for traffic light scheduling : Application to real urban areas	PSO	maximiser le nombre de véhicules qui atteignent leurs destinations, minimiser le temps de trajet global de tous les véhicules	Une seule intersection	microscopic traffic simulator
[5]	A swarm intelligent method for traffic light scheduling : application to real urban traffic networks	PSO	Minimiser le temps d'attente	1 intersection	Simulateur VIS-SIM

## 2.5. COLONIE D'ABEILLES ARTIFICIELLES (ABC)

### 2.5.1. PRINCIPE

L'algorithme ABC (Artificial Bee Colony) est une approche révolutionnaire développée par Karaboga et al [56]. L'ABC possède de bonnes capacités d'exploration pour trouver les poids optimaux pendant le processus de formation [57]. L'algorithme ABC fonctionne sur le principe du comportement de butinage des abeilles à la recherche de nourriture de qualité. Chaque cycle de recherche comprend trois étapes : l'envoi d'abeilles employées sur la source de nourriture pour mesurer la quantité de nectar, la sélection de la source de nourriture par les observateurs une fois l'information partagée par les abeilles employées, et l'envoi d'éclaireurs pour découvrir une nouvelle source de nourriture[58].

### 2.5.2. FONCTIONNEMENT D'ALGORITHME

- Initialisation de la population avec des solutions aléatoires. Réglage des paramètres comme le nombre d'abeilles scout, le nombre de sites choisis, le nombre des sites les plus sélectionnés, le nombre d'abeilles utilisées pour les meilleurs sites, nombre d'abeilles utilisées pour les autres sites sélectionnés, et la taille initiale de la pièce, qui comprend le site et ses environs.
- Évaluer la fonction de fitness de la population.
- Créer la nouvelle population ;
- Sélectionner les sites pour la recherche du voisinage et les abeilles possédant les valeurs de fonction de fitness les plus élevées sont choisies ;
- Recruter des abeilles pour les sites sélectionnés (plus d'abeilles pour les meilleurs sites, et calculer la fonction de fitness. [59]

### 2.5.3. TRAVAUX CONNEXES

Dans cette référence [62] mentionnée, un modèle de réseau de circulation urbaine centralisé est proposé pour résoudre le problème de planification des feux de signalisation. Un algorithme amélioré de colonie d'abeilles artificielles (IABC) est utilisé, avec une stratégie de génération de solutions et des opérateurs de recherche locale.

Les résultats des expériences montrent que l'ABC est compétitif pour résoudre ce problème complexe.

Risikat et Kabir [63] présentent un système de contrôle adaptatif du trafic qui optimise la synchronisation des feux verts pour minimiser le temps d'attente moyen. Les résultats montrent une amélioration significative de la performance du système avec une réduction du temps d'attente et une évacuation efficace des véhicules.

TABLE 2.3 – Travaux Connexes sur les Algorithmes de gestion de trafic basé sur ABC.

Ref	Titre de papier	Algorithme utilisé	Paramètres à optimiser	Nombre d'intersection	Implémentation
[62]	Improved artificial bee colony algorithm for solving urban traffic light scheduling problem	ABC	minimiser le temps d'attente total	Une seule intersection	Simulateur implémenté
[63]	Development and Simulation of Adaptive Traffic Light Controller Using Artificial Bee Colony Algorithm	ABC	Minimiser le temps d'attente moyen	1 intersection	Simulateur implémenté

## 2.6. RECHERCHE DE COUCOU (CS)

### 2.6.1. PRINCIPE

La recherche de coucou (CS) est un algorithme métaheuristique récemment développé par Yang et Deb [64], inspiré par le parasitisme naturel de la race des espèces de coucou voir figure 2.3. Pour comprendre son principe de fonctionnement, considérez que chaque oiseau prend un œuf à la fois et le jette dans un nid aléatoire qui représente une solution unique. Le nid avec des œufs de haute qualité sera transféré à la prochaine génération. Le nombre de nids hôtes est fixé et l'œuf pondu par le coucou est découvert par l'oiseau hôte. Dans cette situation, l'oiseau hôte se débarrasse de l'œuf ou abandonne le nid en développant un nouveau nid [65].





FIGURE 2.3 – Oiseau de coucou.

### 2.6.2. FONCTIONNEMENT

- Les espèces hôtes de coucou forment un système de course aux armements où les œufs de coucou peuvent être découverts et abandonnés avec une probabilité  $p_a$ .
- Ce système avec  $n$  œufs, et coder la position d'un œuf comme un vecteur solution  $x_i$  à un problème d'optimisation. La similarité de deux œufs (solutions  $x_i$  et  $x_j$ ) peut être grossièrement mesurée par leur différence ( $x_j - x_i$ ). Ainsi, la position à l'itération  $t$  peut être mise à jour par :

$$x_i^{t+1} = x_i^t + \alpha s * H(p_a - e) * (x_j^t - x_k^t) \quad (2.6)$$

Où  $x_j^t$  et  $x_k^t$  sont deux solutions différentes sélectionnées aléatoirement par permutation aléatoire,  $H(u)$  est une fonction de Heaviside,  $e$  est un nombre aléatoire tiré d'une distribution uniforme et  $s$  est la taille du pas. Ici, le  $*$  signifie le produit par entrée [66].

D'autre part, la marche aléatoire globale est réalisée en utilisant les vols de Lévy [66] :

$$x_i^{t+1} = x_i^t + \alpha L(s, \lambda) \quad (2.7)$$

$$L(s, \lambda) = \frac{\lambda \Gamma(\lambda) \sin(\pi\lambda/2)}{\pi} \frac{1}{S^{1+\lambda}} \quad (2.8)$$

Ici  $\alpha > 0$  est le facteur d'échelle de taille de pas, qui doit être lié aux échelles du problème d'intérêt. Dans la plupart des cas, nous pouvons utiliser  $\alpha = O(L/10)$ , où  $L$  est l'échelle caractéristique du problème d'intérêt, alors que

dans certains cas  $\alpha = O(L/100)$  peut être plus efficace et éviter de voler trop loin. Bien entendu, la valeur de  $\alpha$  dans ces deux équations de mise à jour peut être différente, conduisant ainsi à deux paramètres différents,  $\alpha_1$  et  $\alpha_2$ . Ici, nous utilisons  $\alpha_1 = \alpha_2 = \alpha$  pour plus de simplicité.

La fonction  $\Gamma$  n'est qu'une constante pour un  $\lambda$  donné. Par exemple, lorsque  $\lambda = 1$ , on a  $\Gamma(1 + \lambda) = 1$ ,  $\Gamma((1 + \lambda)/2) = 1$

### 2.6.3. TRAVAUX CONNEXES

Araghi et Khosravi [67] présente une méthode pour concevoir un contrôleur de signalisation de trafic efficace pour une intersection isolée en utilisant une technique appelée recherche du coucou. L'étude vise à optimiser les performances de la signalisation du trafic en utilisant un système ANFIS (Adaptive Neuro-Fuzzy Inference System). Les auteurs ont évalués les performances d'ANFIS en utilisant CS. L'amélioration des performances de l'ANFIS-CS a été obtenue par rapport au contrôleur à temps fixe. Araghi et Khosravi [68] proposent une autre approche en utilisant l'algorithme de recherche du coucou comme méthode d'optimisation pour ajuster les paramètres des contrôleurs intelligents dans le contexte de la congestion du trafic urbain. Les résultats montrent que les contrôleurs intelligents entraînés avec cette méthode surpassent les contrôleurs à temps fixe, avec des performances moyennes améliorées de 44%, 39% et 35% pour les contrôleurs NN, ANFIS et Q-learning respectivement.

TABLE 2.4 – Travaux connexes sur les algorithmes de gestion de trafic basé sur CS.

Ref	Titre de papier	Algorithme utilisé	Paramètres à optimiser	Nombre d'intersection	Implémentation
[67]	Design of an optimal ANFIS traffic signal controller by using cuckoo search for an isolated intersection	cuckoo search	minimiser le temps d'attente	Une seule intersection	Simulateur implémenté
[68]	Intelligent cuckoo search optimized traffic signal controllers for multi-intersection network	cuckoo search	Minimiser le temps d'attente moyen	1 intersection	Simulateur implémenté

## 2.7. ALGORITHME DE CHAUVÉ-SOURIS (BAT ALGORITHM (BA))

### 2.7.1. PRINCIPE

L'algorithme de chauve-souris (BA), initialement développé par Xin-she yang en 2010, s'inspire de l'écholocation des micro-chauves-souris [69]. Le principe de fonctionnement du BA comprend trois étapes de base : les chauves-souris utilisent l'écholocation pour détecter la distance qui sépare la nourriture de la barrière ; les chauves-souris volent de manière aléatoire avec une intensité sonore et une longueur d'onde variables. Les chauves-souris ajustent automatiquement leur longueur d'onde et leur impulsion en fonction de la proximité de la nourriture/proie [70].

### 2.7.2. FONCTIONNEMENT

- Tout d'abord le démarrage position, vitesse, et la fréquence sont initialisées pour chaque chauve-souris [71]. Pour chaque pas de temps  $t$ , étant  $T$  la limite des itérations, le mouvement des chauves-souris virtuelles est spécifié en mettant à jour leur vitesse et leur position au moyen des équations 2.9, 2.10 et 2.11 comme suit :

$$f_i = f_{min} + (f_{max} - f_{min})\beta \quad (2.9)$$

$$v_i^t = v_i^{t-1} + [x_i^{t-1} - x_*]f_i \quad (2.10)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (2.11)$$

- Où  $\beta \in [0, 1]$  est un vecteur aléatoire tiré d'une distribution uniforme,  $f_i$  dénote la fréquence de chaque chauve-souris, ici,  $x_*$  est le meilleur global actuel solution(localisation) qui est localisée après comparaison de toutes les solutions parmi toutes les  $n$  chauves-souris, à chaque itération [71]. Après la mise à jour de la position des chauves-souris, un nombre aléatoire est généré, si le nombre aléatoire est supérieur au taux d'émission d'impulsions  $r_i$ , une nouvelle position sera générée autour des meilleures solutions actuelles, et elle peut être représentée par l'équation :

$$x_{new} = x_{old} + \epsilon A^t \quad (2.12)$$

- Où,  $\varepsilon \in [-1, 1]$ , est un nombre aléatoire, tandis que  $A^t$  est le volume sonore moyen de toutes les chauves-souris à l'itération actuelle. De plus, la sonie  $A^t$  et le taux d'émission d'impulsions  $r_i$  seront mis à jour et une solution sera acceptée si un nombre aléatoire est inférieur à la sonie  $A_i$  et  $f(x_i) < f(x_*)$ .  $A_i$  et  $r_i$  sont mis à jour par :

$$A_i^{t+1} = aA_i^t, r_i^{t+1} = r_i^0[1 - \exp(-\lambda t)] \quad (2.13)$$

Où  $a$  et  $\lambda$  sont des constantes, l'algorithme itère jusqu'à ce que les critères de terminaison soient remplis [71].

### 2.7.3. TRAVAUX CONNEXES

Srivastava, Sahana[72] ont utilisé le BA pour déterminer le temps d'attente à un feu de circulation pour le modèle microscopique discret. L'étude était basée sur 12 nœuds et quatre intersections. Les résultats ont été comparés avec les AG. Des performances relativement plus élevées ont été obtenues pour l'algorithme BA par rapport à AG. Jintamuttha et al. [73] ont réalisé une simulation expérimentale pour le temps de vert d'une intersection pour dix cycles par course. Les résultats de l'expérience ont été optimisés à l'aide de BA. La longueur moyenne de la file d'attente et le temps d'attente se sont améliorés grâce à l'optimisation.

TABLE 2.5 – Travaux connexes sur les algorithmes de gestion de trafic basé sur BA.

Ref	Titre de papier	Algorithme utilisé	Paramètres à optimiser	Nombre d'intersection	Implémentation
[74]	Bat Algorithm-Based Traffic Signal Optimization Problem	Bat algorithm	minimiser le temps d'attente	4 intersections	Simulateur implémenté
[75]	Dynamic traffic light timing optimization model using bat algorithm	Bat algorithm	Minimiser le temps de détérioration	1 intersection	Simulateur SUMO

## 2.8. SYSTÈME IMMUNITAIRE ARTIFICIEL (AIS)

### 2.8.1. PRINCIPE

L'algorithme de réseau immunitaire (INA) ou système immunitaire artificiel (AIS) est un autre algorithme d'optimisation utile récemment pratiqué pour les

problèmes d'optimisation du contrôle du signal. le mécanisme de fonctionnement de cet algorithme est inspiré du système immunitaire biologique. Les cellules immunitaires possèdent des récepteurs capables de détecter des agents pathogènes nocifs et d'activer des anticorps pour les combattre, conduisant à leur élimination [76].

### 2.8.2. FONCTIONNEMENT

Les principales caractéristiques de la théorie de la sélection clonale du système immunitaire artificiel sont les suivantes :

- Les nouvelles cellules sont des doubles de ces parents (clone) qui sont chargés de la hauteur du mécanisme de mutation rapide (hyper-mutation somatique),
- Élimination du nouveau lymphocyte différent en faisant des récepteurs auto-réactifs,
- La prolifération et la différenciation se font par contact des cellules adultes avec l'antigène.
- La persistance de clones illégaux, résistants à l'élimination précoce par l'auto-antigène, comme base de la maladie auto-immune [77].

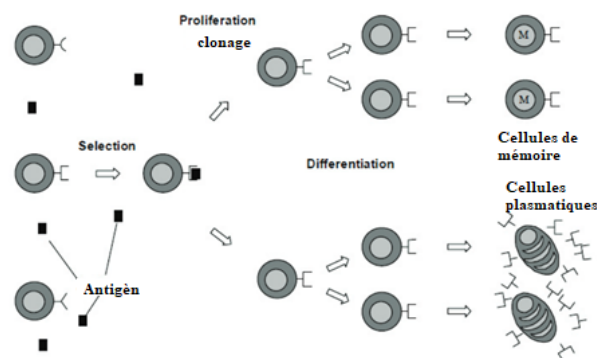


FIGURE 2.4 – Principe général du système immunitaire artificiel.

### 2.8.3. TRAVAUX CONNEXES

Le tableau 2.7 présente les références [78], [79], [80] et [81] qui utilisent l'algorithme AIS pour minimiser le nombre de voitures en attente.

TABLE 2.6 – Travaux connexes sur les algorithmes de gestion de trafic basé sur AIS.

Ref	Titre de papier	Algorithme utilisé	Paramètres à optimiser	Nombre d'intersec-tion	Implémentation
[78]	Traffic Control at Intersections Using Artificial Immune System Approach	L'algorithme de réseau immuni-taire	minimiser le nombre de voiture	une seule intersec-tion	Simulateur im-plémenté
[79]	Multi-agent immune networks to control interrupted flow at signalized intersections	L'algorithme de réseau immuni-taire	Minimiser le nombre de voiture en attente	une seule intersec-tion	Simulateur VIS-SIM
[80]	An artificial immune network to control traffic at a single intersection	L'algorithme de réseau immuni-taire	minimiser le temps d'attente moyen des véhicules	une seule intersec-tion	Simulateur im-plémenté
[81]	Micro Artificial Immune System for Traffic Light Control	L'algorithme de réseau immuni-taire	optimisation des flux de véhicules	une seule intersec-tion	Simulateur SUMO

## 2.9. OPTIMISEUR DES LOUPS GRIS (GWO)

### 2.9.1. PRINCIPE

Gray Wolf Optimizer (GWO) est une nouvelle technique métaheuristique récemment proposée par Mirjalili en 2014 [82]. GWO s'inspire de la hiérarchie sociale et du comportement de chasse des loups gris. Dans l'optimisation GWO, les loups représentent un ensemble de solutions candidates. Le cycle de chasse dans le GWO commence par l'acquisition d'une population aléatoire de solutions candidates (loups) suivie par l'identification des emplacements des proies optimales en utilisant un processus cyclique. GWO présente plusieurs avantages par rapport aux approches évolutives, à la programmation et à la mise en œuvre faciles, à la simplicité de l'algorithme, à un besoin de paramètres spécifiques à l'algorithme et à une complexité de calcul plus faible [83].

### 2.9.2. FONCTIONNEMENT

- Chasser et approcher la proie (Suivre) : les loups gris ont la capacité de reconnaître l'emplacement des proies et les encercler. La chasse est généralement guidée par l'alpha. La beta et delta peuvent également participer occasionnellement à la chasse. Pour simuler mathématiquement le comportement de chasse des loups gris, on suppose que l'alpha (meilleure solution candidate) bêta et delta ont une meilleure connaissance sur l'emplacement potentiel des proies.
- Encercler et harceler la proie (Poursuivre) : les loups gris encerclent leurs proies pendant la chasse. Afin de modéliser mathématiquement le comportement d'encerclement [82].
- Attaque vers la proie [82] : les loups gris terminent la chasse en attaquant la proie lorsqu'elle s'arrête de bouger. Avec les opérateurs proposés jusqu'à présent, l'algorithme GWO permet ses agents de recherche pour mettre à jour leur position en fonction de l'emplacement de l'alpha, le bêta et le delta ; et attaque vers la proie. Cependant, l'algorithme GWO est sujet à la stagnation des solutions locales avec ces opérateurs. Il est vrai que le mécanisme d'encerclement proposé montre l'exploration dans une certaine mesure [82].

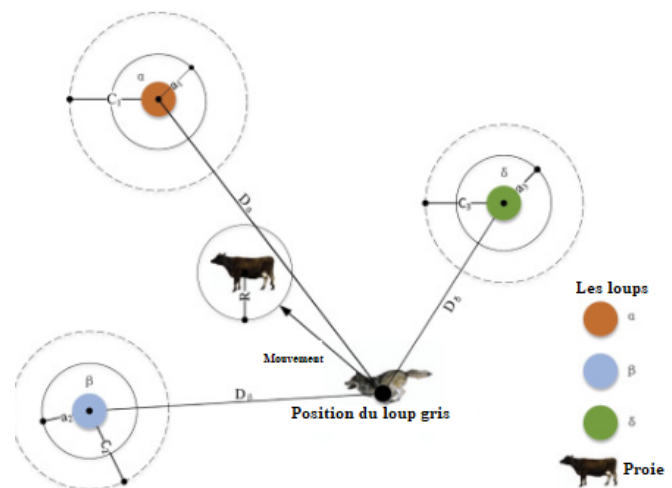


FIGURE 2.5 – Les loups gris (GWO).

### 2.9.3. TRAVAUX CONNEXES

Ces dernières années, GWO a été de plus en plus utilisé dans diverses disciplines. Cependant, les études sur ses applications en matière de transport et d'ingénierie du trafic en général et le contrôle du trafic et l'optimisation en particulier sont très peu nombreuses. Le tableau 2.8 présente les travaux de GWO.

TABLE 2.7 – Travaux connexes sur les algorithmes de gestion de trafic basé sur GWO.

Ref	Titre de papier	Algorithme utilisé	Paramètres à optimiser	Nombre d'intersection	Implémentation
[84]	A hybrid algorithm based on GWO and GOA for cycle traffic light timing optimization	Grey Wolves Optimizer	Minimiser le temps d'attente, maximiser le nombre de véhicule arrivés à leurs destination	Une seule intersection	Simulateur SUMO
[85]	Grey wolf optimizer based fuzzy-PI active queue management design for network congestion avoidance	Grey Wolves Optimizer	la gestion active des files d'attente des véhicules	1 intersection	Simulateur implémenté

## 2.10. RECHERCHE D'HARMONIE (HS)

### 2.10.1. PRINCIPE

L'algorithme de recherche d'harmonie métaheuristique (HS) simule le processus d'improvisation musicale naturelle où les musiciens visent à atteindre un état d'harmonie presque parfait [86]. Dans l'algorithme HS, la population de solution candidate est connue sous le nom de mémoire harmonie (HM), où chaque solution dans l'espace de solution est appelée «harmonie», qui appartient au vecteur de dimension «N».



### 2.10.2. FONCTIONNEMENT

- Initialiser l'algorithme en spécifiant des paramètres, initialisez la mémoire Harmony (HM) et triez pour trouver la pire harmonie,  $X_{worst}$
- Générer une nouvelle harmonie,  $X_{new}$ , mettez à jour HM si  $X_{new}$  est meilleur que  $X_{worst}$  [86], [89].

### 2.10.3. TRAVAUX CONNEXES

Le tableau 2.9 présente les travaux de la recherche d'harmonie.

TABLE 2.8 – Travaux connexes sur les algorithmes de gestion de trafic basé sur HS.

Ref	Titre de papier	Algorithme utilisé	Paramètres à optimiser	Nombre d'intersec-tion	Implémentation
[87]	A hybrid harmony search and TRANSYT hill climbing algorithm for signalized stochastic equilibrium transportation networks	Recherche d'harmonie	network performance index (PI)	6 intersec-tions	Simulateur TRANSYT
[88]	Optimizing urban traffic light scheduling problem using harmony search with ensemble of local search	Recherche d'harmonie	Minimiser le temps d'attente total	3 intersec-tions	Simulateur implémenté

### 2.11. CONCLUSION

Un bon contrôle des feux de circulation sur le réseau routier peut résoudre les embouteillages dans les villes. Dans ce chapitre, nous avons étudié quelques travaux connexes ayant abordé les techniques d'optimisation de la synchronisation des feux de circulation basées sur des métaheuristiques. Nous avons traité les méta-heuristiques les plus étudiés qui sont PSO, ACO, GA, GWO, CS, HS, AIS, BA, ABC.

Vue que la majorité des techniques réel de synchronisations des feux de signalisation utilise l'aspect statique, cette dernière ignore le contrôle de feu de circulation. Donc il est suggéré d'optimiser le problème de contrôle de trafic avec les algorithmes d'optimisation méta-heuristique afin de minimiser le nombre des véhicules en attente .

Notre proposition, qui fera l'objet du chapitre suivant, sera compris deux approche principaux basées sur l'algorithme génétique et l'algorithme de recherche symbiotique discrete afin de minimiser le nombre des automobiles en attente.

## APPROCHES PROPOSÉES

### Sommaire

---

<b>3.1</b>	<b>Introduction</b> . . . . .	<b>59</b>
<b>3.2</b>	<b>Optimisation et méthodes méta-heuristiques</b> . . . . .	<b>60</b>
3.2.1	Définition de l'optimisation . . . . .	60
3.2.2	Problème d'optimisation . . . . .	61
3.2.3	Méthodes classiques v.s. Méthodes métaheuristiques	62
<b>3.3</b>	<b>Approche 1 : Description de la première stratégie proposée (DJ-GA)</b> . . . . .	<b>65</b>
3.3.1	Niveau1 : Détermination des feux de signalisation critiques . . . . .	67
3.3.2	Description de la fonction objective . . . . .	68
3.3.3	Niveau 2 : Optimisation des feux de signalisation par l'Algorithme Génétique . . . . .	70
3.3.4	Pourquoi les algorithmes génétiques fonctionnent-ils? . . . . .	73
3.3.5	Adaptation de l'Algorithme Génétique . . . . .	74
<b>3.4</b>	<b>Approche 2 : Description de la stratégie proposée(ACO-DSOS)</b> . . . . .	<b>78</b>
3.4.1	Niveau 1 : Recherche des feux critiques par l'algorithme de colonie des fourmies (ACO) . . . . .	79
3.4.2	Niveau02 :Optimisation des feux de signalisation par l'Algorithme Discret Symbiotic Organism Search	82
<b>3.5</b>	<b>Conclusion</b> . . . . .	<b>89</b>

---

### 3.1. INTRODUCTION

Les feux de signalisation jouent un rôle essentiel dans la régulation de la circulation en contrôlant les flux de véhicules aux intersections. Ils sont des dispositifs essentiels pour réguler la circulation. Cependant, une mauvaise coordination des feux de signalisation peut entraîner des temps d'attente excessifs, des congestions de trafic et une augmentation des émissions de gaz à effet de serre.

L'optimisation des feux de signalisation est un problème qui consiste à trouver les meilleures configurations et séquences de feux afin de minimiser les temps d'attente des véhicules et d'améliorer la fluidité du trafic. Cela permet d'offrir aux usagers de la route des déplacements plus fluides, plus sûrs et plus efficaces.

Cependant, l'optimisation des feux de signalisation est un problème complexe en raison de la variété des conditions de trafic, des contraintes de temps, des priorités de circulation et des caractéristiques spécifiques à chaque intersection. De plus, les réseaux routiers sont souvent dynamiques, avec des fluctuations constantes des volumes de trafic et des demandes changeantes. Pour résoudre ces défis, diverses approches d'optimisation sont utilisées, ces approches comprennent des méthodes exactes et approchées. Les méthodes exactes, telles que la programmation linéaire [94], la programmation dynamique [90] [91] et la méthode Branch and Bound [92] [93]...etc, visent à trouver la solution optimale en explorant systématiquement toutes les possibilités. Cependant, ces méthodes peuvent être limitées en termes de temps de calcul pour des problèmes de grande taille. D'autre part, les méthodes approchées sont des approches plus pragmatiques qui fournissent des solutions réalisables, mais sans garantie d'optimalité. Ces méthodes sont souvent utilisées pour des problèmes de grande taille ou pour des problèmes où l'optimalité est difficile à déterminer. Les méthodes approchées cherchent à trouver des solutions de qualité acceptable dans un temps raisonnable.

Les problèmes d'optimisation des feux de signalisation sont NP-difficiles, ce qui signifie qu'ils sont intrinsèquement difficiles à résoudre en temps polynomial. Trouver une solution exacte pour un réseau routier de taille significative nécessite souvent une exploration exhaustive de toutes les combinaisons possibles de séquences de feux, ce qui entraîne une complexité exponentielle.

Dans cette étude, nous nous pencherons sur l'optimisation des feux de signalisation dont on a appliqué les méta-heuristiques ci-dessous pour fournir une optimisation efficace.

Notre objectif principal est proposé et d'implémenter deux approches afin de minimiser les feux de signalisations dans des intersections critiques où un grand nombre de véhicules attendent.

Le présent chapitre permet de décrire nos stratégies proposées, d'expliquer les démarches, détailler les différentes phases ainsi qu'à la description des algorithmes nécessaires pour leur mise en œuvre.

### **3.2. OPTIMISATION ET MÉTHODES MÉTA-HEURISTIQUES**

#### **3.2.1. DÉFINITION DE L'OPTIMISATION**

L'optimisation est le fait d'obtenir le meilleur résultat dans des circonstances données. Dans tout domaine d'ingénierie, les ingénieurs doivent prendre beaucoup de décisions technologiques et de gestion. Le but ultime de toutes ces décisions est soit minimiser l'effort requis ou maximiser le bénéfice souhaité.

Cependant, l'effort et le bénéfice dans toute situation concrète peut se modéliser en une fonction coût de certaines variables de décision. L'optimisation peut être définie comme le processus de trouver les variables qui donnent le maximum ou minimum de cette fonction, sans ou avec contraintes[111].

### 3.2.2. PROBLÈME D'OPTIMISATION

D'un point de vue mathématique, le problème se formule de la façon suivante :

- problème sans contraintes :

$$\min_{x \in \mathbb{R}^{n_x}} J(x) \quad (3.1)$$

- problème avec contraintes :

$$\min_{x \in \mathcal{C}} J(x) \quad (3.2)$$

où  $x = [x_1 \ x_2 \ \dots \ x_{n_x}]^T$  est le vecteur des  $n_x$  variables de décision.

$J(x) = [J_1(x) \ J_2(x) \ \dots \ J_M(x)]^T$  est le vecteur des  $M$  fonctions coût.

Quand  $M > 1$ , le problème est dit d'optimisation multi-objective.

La recherche d'un minimum de  $J(x)$  est équivalent à la recherche d'un maximum de  $-J$ , et vis-versa[111].

L'optimisation est souvent modélisée sous la forme (3.2). Les contraintes sont :

- soit de type inégalités avec :

$$\mathcal{C} = \{x \in \mathbb{R} \text{ tels que } :h_l(x) \leq 0, i = 1, \dots, m_l\} \quad (3.3)$$

- soit de type égalité avec :

$$\mathcal{C} = \{x \in \mathbb{R} \text{ tels que } :g_l(x) = 0, i = 1, \dots, m_e\} \quad (3.4)$$

avec  $m_i$  est le nombre des contraintes de type inégalité et  $m_e$  est le nombre de contraintes de type égalité [111].

L'optimisation avec contraintes (3.2) peut se ramener à un problème d'optimisation sans contraintes (3.1) en introduisant une nouvelle fonction coût qui inclue les contraintes de la manière suivante :

$$L(x, \lambda) = J(x) + \sum_{l=1}^{m_i} \lambda_l g_l(x) + \sum_{l=m_e+1}^{m_e+m_i} \lambda_l \max(h_{l-m_e}(x), 0) \quad (3.5)$$

Le terme ajouté à  $J(x)$  est appelé fonction de pénalité,  $L(x, \lambda)$  le lagrangien et  $\lambda_l$  les multiplicateurs de lagrange [111]. Le choix des  $\lambda_l$  n'est pas critique pour l'optimisation, en effet, ces derniers pénalisent plus ou moins la violation des contraintes. Si  $x$  satisfait les contraintes, nous aurons  $L(x, \lambda) = J(x)$  et le problème d'optimisation devient :

$$\min_{x \in \mathbb{R}^{n_x}} L(x, \lambda) \quad (3.6)$$

### 3.2.3. MÉTHODES CLASSIQUES V.S. MÉTHODES MÉTAHEURISTIQUES

L'existence des méthodes d'optimisation peut être attribuée à l'époque de Newton, Lagrange, et Cauchy. L'utilisation du calcul différentiel dans l'optimisation est devenue possible grâce aux contributions de Newton et Leibnitz. Les pionniers qui ont traité la minimisation des fonctionnelles, étaient Bernoulli, Euler, Lagrange, et Weirstrass.

L'optimisation avec contraintes qui implique l'ajout de multiplicateurs inconnus, est devenu connu par le nom de son inventeur, Lagrange. Cauchy était le premier à utiliser la méthode de descente du gradient pour résoudre les problèmes de minimisation sans contrainte. Malgré ces premières contributions, très peu de progrès ont été réalisés.

Il fallait attendre le milieu du vingtième siècle, lorsque les ordinateurs numériques à haute vitesse faisaient la mise en œuvre des procédures d'optimisation et stimulaient les recherches sur de nouvelles méthodes. Les progrès spectaculaires suivis, produisant une littérature considérable sur les techniques d'optimisation. Mais d'un autre côté, les problèmes à optimiser deviennent plus complexes, et les obstacles pour les résoudre plus émergents. Le principal obstacle est que la plupart des problèmes d'optimisation sont NP-difficiles [96]. Par conséquent, les méthodes théoriques connues ne peuvent pas être appliquées sauf peut-être pour certains problèmes de petite taille. D'autres difficultés paraissent quand la fonction coût n'est pas différentiable et/ou multimodale. Par conséquent, l'ensemble des méthodes utilisant les dérivées de la fonction coût ne peut pas être utilisé. Un autre obstacle est quand la fonction coût ne peut être modélisée sous forme analytique mais évaluée que par des simulations.

Dans ces situations, les approches métaheuristiques semblent être le seul moyen pour résoudre les problèmes d'optimisation.

Le terme métaheuristique vient des mots grecs meta (au delà ) et heuriskein (trouver).

Il n'y a pas clairement de consensus sur la définition exacte des heuristiques et des métaheuristiques. Nous allons adopter celles-ci :

- Une heuristique est une technique de résolution spécialisée à un problème. Elle ne garantit pas la qualité de la solution obtenue.
- Une métaheuristique est une heuristique générique qu'il faut adapter à chaque problème.

### 3.2.3.1. CARACTÉRISTIQUES DES MÉTAHEURISTIQUES

Nous citons ici quelques caractéristiques des métaheuristiques

- ▷ Le but visé par les métaheuristiques est d'explorer l'espace de recherche efficacement afin de déterminer des solutions (presque) optimales.



- ▷ Les techniques qui constituent des algorithmes de type métaheuristique vont de la simple procédure de recherche locale à des processus d'apprentissage complexes.
- ▷ Les métaheuristicques sont en général non-déterministes et ne donnent aucune garantie d'optimalité.
- ▷ Les métaheuristicques peuvent être hybridés avec d'autres métaheuristicques ou avec des méthodes classiques.
- ▷ Les métaheuristicques peuvent contenir des mécanismes qui évitent le blocage dans un minimum local[97].

### 3.2.3.2. CLASSIFICATION DES MÉTAHEURISTIQUES

Les métaheuristicques sont généralement classés en deux grandes familles de méthodes, à savoir, celles dites de trajectoire et celles basées sur les populations.

**Méthodes de trajectoires** elles manipulent une seule solution à la fois et tentent itérativement d'améliorer cette solution. Elles construisent une trajectoire dans l'espace des solutions en tentant de se diriger vers des solutions optimales. Parmi les exemples, nous citons : Le recuit simulé [98] et la recherche tabou [99].

**Méthodes basées sur une population** En tout temps on dispose d'une base de plusieurs solutions, appelée population. Ces métaheuristicques s'inspirent de phénomènes naturels ou biologiques. Nous citons comme exemples les algorithmes génétiques [100] et les algorithmes des fourmis [101] s'inspirent respectivement de la théorie de l'évolution et du comportement de fourmis à la recherche de nourriture.

### 3.3. APPROCHE 1 : DESCRIPTION DE LA PREMIÈRE STRATÉGIE PROPOSÉE (DJ-GA)

La nécessité des services de transport et le développement des réseaux routiers ont émergé avec l'avancement de la civilisation. La croissance démographique et l'urbanisation rapide ont entraîné une augmentation significative du nombre de véhicules circulant dans les zones urbaines, ce qui a entraîné une saturation des routes et des problèmes de congestion. Ces problèmes ont un impact négatif sur l'efficacité des déplacements, la qualité de l'air, la consommation de carburant et la qualité de vie des habitants.

Dans le but de réduire les temps d'attente des véhicules, une solution a été développée en utilisant un modèle à deux niveaux. Le premier niveau vise à identifier les intersections qui génèrent un nombre important de véhicules en attente, c'est-à-dire à optimiser la sélection des intersections sur lesquelles l'algorithme DIJKSTRA sera appliqué.

La sélection des intersections à traiter par l'algorithme DIJKSTRA est basée sur des critères tels que le nombre de véhicules en attente, les heures de pointe, la densité de trafic, etc. Cela permet de concentrer les efforts d'optimisation sur les zones les plus problématiques.

Le deuxième niveau de notre approche vise à optimiser les feux de signalisation à l'aide de l'algorithme génétique (GA).

Dans le contexte de l'optimisation des feux de signalisation, l'algorithme génétique est utilisé pour trouver les meilleurs paramètres de réglage des feux, (les durées vertes et rouges) afin de minimiser les temps d'attente des véhicules et d'optimiser la fluidité du trafic.

La Figure 3.1 illustre cette approche en montrant le flux de travail de notre modèle.

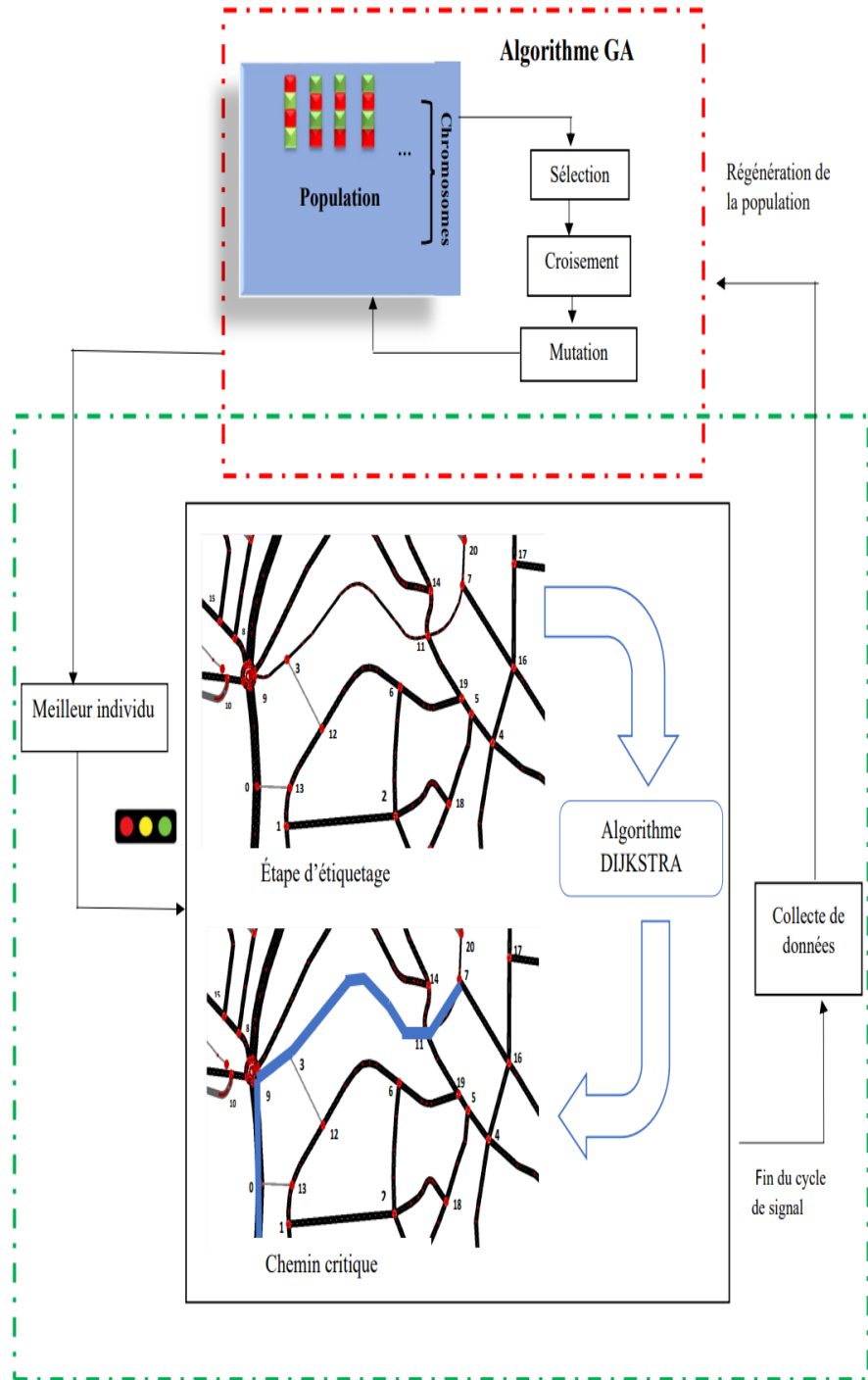


FIGURE 3.1 – Architecture générale de l'approche DJ-GA.

### 3.3.1. NIVEAU1 : DÉTERMINATION DES FEUX DE SIGNALISATION CRITIQUES

La première étape à effectuer consiste à déterminer uniquement les feux de signalisation qui génèrent un nombre important des véhicules en attente en utilisant l'algorithme DIJKSTRA pour trouver le chemin critique. Avant d'appliquer l'algorithme DIJKSTRA [102], en nous inspirant du protocole OSPF (Open Shortest Path First)[103], chaque lien est étiqueté avec l'inverse de la valeur obtenue à l'aide de la formule (3.7) mentionnée ci-dessous afin d'obtenir un chemin plus encombré . Enfin, les feux de signalisation qui entraînent l'attente des véhicules sont déterminés, comme illustré dans la figure 3.2.

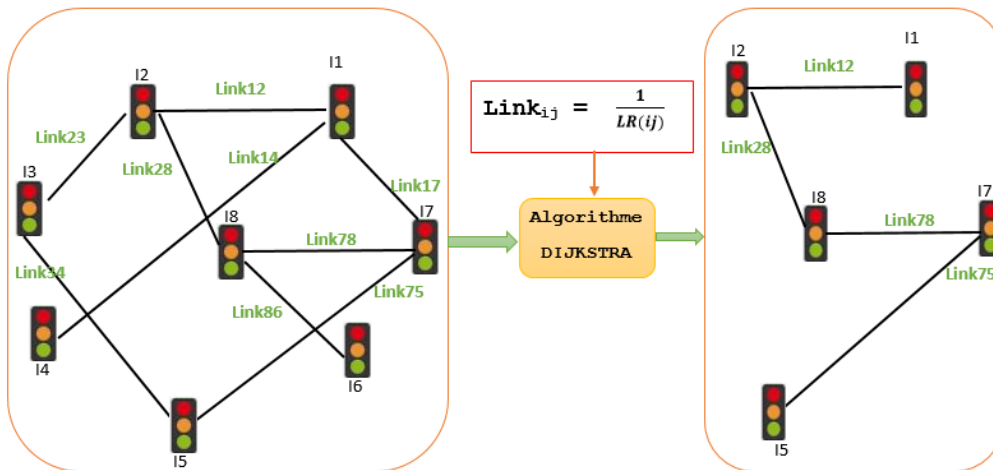


FIGURE 3.2 – Exemple de la première stratégie.

Nous avons représenté la région sous la forme d'un graphe qui comprend des nœuds représentant les feux de signalisation et des arêtes représentant les voies entre les feux. En analysant cette figure 3.2, nous identifions les différents chemins disponibles dans la ville en spécifiant le point de départ et le point de destination.

Pour déterminer le chemin le plus encombré, nous utilisons l'algorithme de Dijkstra qui exploite la connaissance du réseau routier de la ville.

## Approches Proposées

---

Cet algorithme prend en entrée le graphe de la ville et construit progressivement un sous-graphe représentant le chemin critique, c'est-à-dire l'ensemble des feux de signalisation qui génèrent un temps d'attente élevé. Cette étape consiste à trouver un chemin avec une forte charge de trafic en utilisant l'algorithme de Dijkstra, qui fait partie du prétraitement de notre approche.

---

**Algorithme 1** : Pseudo-code de niveau01 (application DIJKSTRA sur les feux de signalisation)

---

- **Input** : MP : Matrice de pondération des liens ;
  - **Output** : Chemin[] : tableau contenant les feux critiques ;
  - Lire (feux vert, feux rouge) pour chaque nœud ;
  - Calculer LR(ij) pour chaque link voir l'équation (3.7) ;
  - Chemin=Dijkstra (MP) ;
  - return Chemin ;
  - Fin.
- 

### 3.3.2. DESCRIPTION DE LA FONCTION OBJECTIVE

Les grandes villes sont confrontées à une augmentation significative de la demande du trafic. La congestion du réseau routier urbain a été un problème pour la plupart des villes du monde. Les indices de performance du problème des feux de circulation sont généralement le nombre de véhicule en attente. L'objectif est de prédire le flux de circulation et d'optimiser la synchronisation des feux de circulation de manière à réduire la congestion du trafic.

Pour notre système, certains types de données du trafic sont nécessaires à l'optimisation, notamment la longueur de la file d'attente, le nombre de véhicule libérée, le temps de feux vert et le temps de feux rouge.

## Approches Proposées

---

La fonction objective est exprimée par la formule suivante :

$$LR(K) = \text{Min}\left\{\frac{LR_{jl}^i(K-1) + \sum_{i=1}^l (q_{jl}^i(K) - \lambda_{jl}^i) * g^i(K)}{C(K)} + \frac{LR_{jl}^i(K-1) + \sum_{i=1}^l q_{jl}^i(K) * R^i(K)}{M(K)}\right\} \quad (3.7)$$

TABLE 3.1 – Paramètres de la fonction objectif.

Element	Description
LR(K)	le nombre de véhicule en attente en feux vert et feut rouge(veh/s).
K	le nombre de cycle.
j	les directions 1, 2, 3, 4 expliquent respectivement l'est, l'ouest, nord, sud de l'intersection.
L	présente la voie (1, 2, 3,4).
l	présente le nombre de phase (1, 2, 3, 4).
$LR_{jl}^i(K-1)$	le nombre des véhicules en attente de cycle K-1 (veh/s).
$q_{jl}^i(K)$	arrivé des véhicules dans la phase i du cycle K (veh/s).
$R^i(K)$	temps de feux rouge dans la phase i du cycle K.
$g^i(K)$	temps de feux vert dans la phase i du cycle K.
$C(K)$	la somme du temps de feux vert du cycle K (S).
$M(K)$	la somme du temps de feux rouge du cycle K (S).
$\lambda_{jl}^i$	les véhicules libérés de la voie L du cycle K (veh/s).

### 3.3.2.1. DESCRIPTION DÉTAILLÉE DES PARAMÈTRES DE L'ÉQUATION (3.7)

1.  $LR_{jl}^0(0)$  : le nombre des vehicules à l'instant 0.
2.  $LR_{jl}^0(K) = LR_{jl}^i(K-1)$  : le nombre des véhicules en attente initiale de cycle K égale à celle de cycle K-1.
3.  $\sum_{i=1}^l g^i(K) = C(K)$  : Le C(K) est égal à la somme des feux vert de chaque phase.
4. Le temps de feux vert doit être respecté comme suit :  $g_{min}^i \leq g^i(K) \leq g_{max}^i$

5. Le  $C(K)$  doit être satisfait comme suit :  $C_{min} \leq C(K) \leq C_{max}$
6.  $\sum_{i=1}^l R^i(K) = M(K)$  : Le  $M(K)$  est égal à la somme des feux rouge de chaque phase.
7. Le temps de feux rouge doit être satisfait comme suit :  $R_{min}^i \leq R^i(K) \leq R_{max}^i$
8. Le  $M(K)$  doit être satisfait par la formule :  $M_{min} \leq M(K) \leq M_{max}$ .

### 3.3.3. NIVEAU 2 : OPTIMISATION DES FEUX DE SIGNALISATION PAR L'ALGORITHME GÉNÉTIQUE

Le noyau du modèle proposé est l'étape d'optimisation des feux de signalisation, cette étape consiste à appliquer une méta-heuristique nommé Genetic Algorithm dans l'objective de minimiser le nombre de voiture en attente.

L'algorithme génétique (AG) est une technique d'optimisation qui s'inspire de la théorie de l'évolution, proposé en 1975 par J. Holland [104], GA ne garantit pas de trouver la solution optimale du problème, cependant il existe des preuves empiriques que les solutions se situent entre des niveaux acceptables, dans un temps compétitif avec le reste des algorithmes d'optimisation combinatoire, c'est-à-dire le recuit simulé, les méthodes de recherche séquentielle, l'hyper-escalade, etc. Burjorjee [105] a proposé une explication de la remarquable capacité d'adaptation des GA. En outre, Burjorjee [105] présente des preuves qui suggèrent fortement que les GA peuvent mettre en œuvre l'hypercalcul de manière extraordinairement efficace pour les problèmes d'optimisation complexes. De plus, les GA ne font aucune hypothèse sur l'espace de recherche du problème d'optimisation. Ce sont quelques-unes des raisons pour lesquelles les GA ont été appliqués pour résoudre un large éventail de problèmes d'ingénierie et d'optimisation scientifique[106].

### 3.3.3.1. FONCTIONNEMENT DES ALGORITHMES GÉNÉTIQUES

Les algorithmes génétiques sont des modèles informatiques basés sur les mécanismes de la sélection naturelle et la théorie de l'évolution. Une application principale des algorithmes génétiques a été dans le domaine de l'optimisation des fonctions. Les algorithmes génétiques sont capables de gérer à la fois les problèmes d'optimisation continue et combinatoire [107],[108].

Par analogie, l'AG maintient une "population" de candidats à la solution pour le problème donné. Des éléments sont tirés au hasard de cette population et autorisés à se "reproduire", en combinant certains aspects des deux solutions parentes. La clé est que la probabilité qu'un élément soit choisi pour se reproduire est basée sur son "fitness", essentiellement une fonction objective liée à la solution [43],[21].

Finalement, les éléments inaptes meurent de la population, pour être remplacés par les descendants de la solution réussie.

Il convient d'abord d'expliquer comment les variables du problème d'optimisation doivent être encodées puis recombinaées. Le fondement théorique de l'AG exige que les variables du problème d'optimisation soient codées dans une chaîne de bits binaires, nombres réels ou caractères. Chaque bit, nombre réel ou caractère de la chaîne dans GA est appelé gène ou paramètre, et ils forment ensemble de chromosome, également appelé individu ou chaîne [47],[22],[23].

Chaque combinaison différente des paramètres dans le chromosome représente une variable différente dans l'espace de recherche du problème d'optimisation.

L'optimisation est effectuée en manipulant la population d'individus en suivant les étapes suivantes :

**Initialisation** Les individus de la population initiale sont affectés à des points du domaine de recherche.



En règle générale, chaque individu est affecté à un point aléatoire dans le domaine de recherche (c'est-à-dire que chaque point du domaine de recherche a la même probabilité d'être choisi).

**Sélection** Les individus de la population sont sélectionnés pour la reproduction. La probabilité de sélection pour chaque individu est généralement la valeur de la fonction objectif. Pour les problèmes de maximisation, les individus avec une valeur de fonction objectif plus grande ont une plus grande probabilité de sélection. Pour les problèmes de minimisation, les individus avec une valeur de fonction objectif plus petite sont favorisés pour la sélection.

**Recombinaison/Crossover** une fois que les individus ont été sélectionnés pour la reproduction, ces "parents" sont appariés et un ou plusieurs "enfants" sont créés à l'aide d'un opérateur de croisement. Le croisement crée des enfants en combinant ou en mélangeant le matériel génétique des deux parents (c'est-à-dire la décision ensembles variables des deux parents sont combinés pour former un nouvel ensemble pour chaque enfant).

**Mutation** l'opérateur de mutation effectue des modifications aléatoires du matériel génétique d'un individu. La mutation modifiera une ou plusieurs des variables de décision de l'individu avec une faible probabilité. La mutation est généralement appliquée aux enfants créés par recombinaison.

**Remplacement** généralement, une taille de population finie fixe est appliquée et un schéma de remplacement est défini pour déterminer quels individus des populations parent et enfant survivront.

L'initialisation est effectuée et les étapes restantes sont répétées jusqu'à ce qu'un critère d'arrêt soit satisfait.

Une condition d'arrêt possible consiste à arrêter la recherche après l'exécution d'un nombre prédéfini d'évaluations de la fonction objectif. L'individu ayant la meilleure valeur de fonction objective est considérée comme la solution optimale produite par l'algorithme[107],[108].

### 3.3.4. POURQUOI LES ALGORITHMES GÉNÉTIQUES FONCTIONNENT-ILS ?

La question se pose naturellement de savoir pourquoi un algorithme tel que décrit dans les sections précédentes aboutira à une recherche d'optimisation efficace. Nous donnons d'abord des justifications qualitatives pour chaque composant de la procédure d'optimisation. Cette explication est suivie d'une explication théorique plus rigoureuse qui met en évidence la puissance intrinsèque de la recherche par algorithme génétique.

#### 3.3.4.1. JUSTIFICATIONS QUALITATIVES DES ÉTAPES D'OPTIMISATION

**La sélection** : En favorisant les meilleures solutions, la procédure de sélection oriente la recherche vers les régions les plus performantes de l'espace de recherche.

**Crossover** : Le croisement effectue un échange d'informations structuré mais aléatoire, conduisant l'algorithme vers de nouveaux points dans l'espace de recherche. Pris en conjonction avec la sélection, l'algorithme est plus susceptible de rechercher des points proches des individus les plus performants. la probabilité de croisement peut être utilisée pour contrôler le taux d'exploration. Régler  $P_c=1$  et toujours effectuer des croisements peut entraîner la perte de bon matériel génétique avant qu'il n'ait été suffisamment exploité. Avec  $P_c < 1$ , il est possible que le matériel génétique passe intact à la génération suivante.

**Mutation** : les avantages de la mutation sont doubles

- La mutation induit des altérations aléatoires du matériel génétique, si l'itération induite est souhaitable en ce sens que la valeur de la fonction objective est améliorée, alors l'individu aura plus de chances de transmettre son matériel génétique grâce à la procédure de sélection. Ainsi, la mutation peut être considérée comme une composante

ascensionnelle de la recherche. En fait, un algorithme de recherche utilisant la mutation seule peut être efficace.

- La sélection, le croisement peuvent entraîner la perte de matériel génétique potentiellement utile (par exemple, pour le codage binaire, un 0 peut ne plus être présent à un bit particulier position chez n'importe quel individu de la population). Alternativement, la population initiale peut manquer de ce matériel génétique. La mutation permet la restauration ou l'introduction de ce matériel génétique dans un nouveau contexte où il peut être trouvé pour améliorer la valeur de la fonction objective [107],[108].

### 3.3.5. ADAPTATION DE L'ALGORITHME GÉNÉTIQUE

Notre approche présente l'adaptation du système de contrôle intelligent des feux de circulation basée sur l'algorithme génétique afin de trouver des programmes cyclables optimaux (ou quasi optimaux) pour les feux de circulation. Le résultat du premier niveau est l'entrée du deuxième niveau, cette dernière permet d'utiliser l'algorithme génétique sur l'ensemble des feux critiques obtenus dans le premier niveau. Cependant, nous avons opté la solution proposée sur plusieurs carrefours dont nous calculons le nombre des véhicules en attente au feu vert et le nombre des véhicules en attente au feu rouge après nous appliquons l'algorithme génétique sur une population de chromosome afin de choisir le meilleur chromosome contenant les meilleurs feux selon la fonction objective décrit ci-dessus. Nous devons générer au hasard des chromosomes au début, Il y a trois opérations principales : la sélection, le croisement et la mutation. La première étape consiste à sélectionner les parents qui ont de bonnes valeurs de fitness pour la reproduction. Ensuite, utilisez ces chromosomes sélectionnés pour effectuer des croisements et des mutations. La méthode de croisement utilisée dans notre approche est le croisement à un point, ce qui signifie que deux parents ont tous deux un seul point de croisement et qu'ils sont échangés par ce point pour recombinaison. L'étape de mutation est utilisée pour la diversité génétique en remplaçant l'un des fragments

de chromosome (gène) dans notre méthode. La probabilité de croisement et de mutation est fixée. Enfin, en utilisant la méthode de la « roulette » pour sélectionner les chromosomes de haute qualité qui ont des valeurs de fitness inférieures dans notre méthode pour la prochaine génération.

### 3.3.5.1. REPRÉSENTATION DU CHROMOSOME

Les chromosomes sont les solutions possibles qui forment l'ensemble de la population dans le processus d'algorithme génétique. Un seul chromosome est utilisé pour coder l'information d'une solution. Dans la gestion de la synchronisation des signaux de signalisation d'optimisation basée sur l'algorithme génétique, un chromosome contient les paramètres de synchronisation des signaux des intersections tels que le temps de cycle qui contient la séquence de phase (temps de feux vert et le temps de feux rouge) car ce sont les paramètres qui ont optimisés dans l'algorithme génétique pour le plan des feux de circulation. Le codage entier est sélectionné pour coder le chromosome. Cela signifie que chaque gène est défini par un feu vert et un feu rouge, La figure 3.3 montre la disposition des chromosomes proposés.

Cycle	Gène 01		Gène 02		Gène 03		Gène 04	
	Phase 01		Phase 02		Phase 03		Phase 04	
Feux	Rouge	Vert	Rouge	Vert	Rouge	Vert	Rouge	Vert
Exemple	40	68	22	52	34	64	19	30

FIGURE 3.3 – Représentation du chromosome.

Par conséquent, le schéma d'optimisation proposé permet d'optimiser le temps de cycle qui contient la séquence de phase (temps de feux rouge et feux vert) de chaque intersection en tenant compte du retard du trafic et de la fluidité.

3.3.5.2. PROCESSUS DE SÉLECTION

La sélection est la première étape de l'algorithme pour refléter le mécanisme génétique. La fonction est de choisir des chromosomes avec une grande capacité d'adaptation parmi la population contemporaine pour se préparer aux étapes suivantes. Le but est de sauver les bons chromosomes et d'éliminer les chromosomes inférieurs et de faire converger l'algorithme. La méthode de sélection choisie est la méthode de roulette cette dernière permet à chaque génération de la population d'avoir les meilleurs chromosomes de la génération précoce à la génération actuelle. Le mode de fonctionnement spécifique est de sélectionner directement les meilleurs chromosomes du groupe actuel vers la génération suivante, de sorte que les meilleurs chromosomes ne puissent pas être détruits par des opérations de croisement et de mutation. De plus, la sélection de roulette peut être étendue pour sélectionner directement certains des membres actuels du groupe à la génération suivante sans croisement.

3.3.5.3. PROCESSUS DE CROISEMENT

Au cours de ce processus, des chromosomes sélectionnés auront la possibilité d'échanger des informations génétiques avec d'autres chromosome pour produire une progéniture via un opérateur de croisement. La stratégie de croisement proposée est un croisement à un points, ce qui augmente la résolution de l'opération de croisement, de sorte que de petits changements sur la solution peuvent être apportés. L'opérateur de croisement est conçu comme un point aléatoire à chaque groupe de paramètres qui sera choisi pour le croisement.

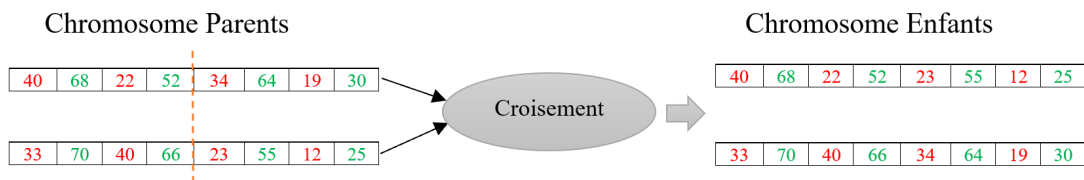


FIGURE 3.4 – *Processus de Croisement.*

3.3.5.4. PROCESSUS DE MUTATION

Le chromosome aura une chance de muter. Cet opérateur évite que la solution ne soit piégée à des maximums locaux en permettant d'ajouter de petits changements de déviation aux gènes sélectionnés. Ensuite, les chromosomes mutés seront impliqués dans le processus d'évolution. À la fin du processus d'évolution, les changements causés par la mutation sont très faibles et la solution peut être affinée pour localiser l'optimum évaluer.

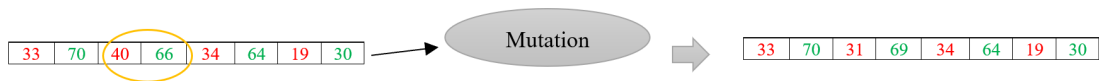


FIGURE 3.5 – Processus de Mutation.

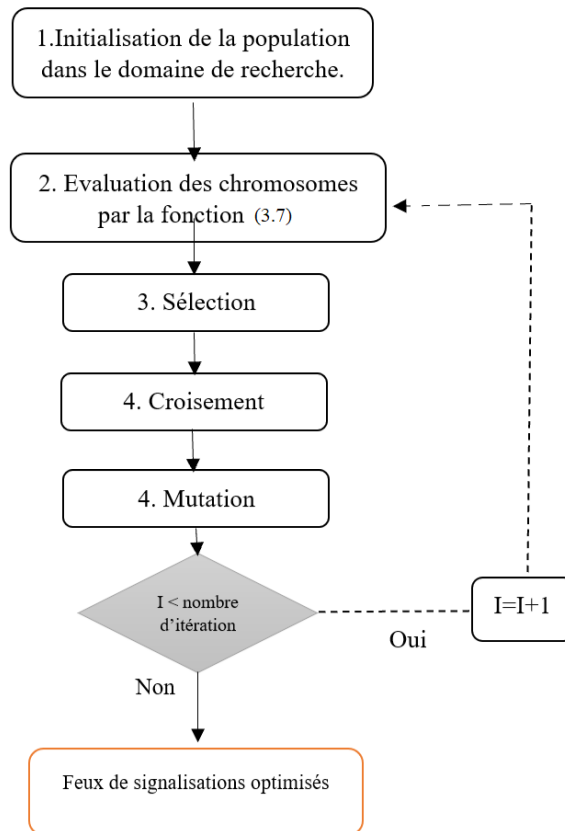


FIGURE 3.6 – L'organigramme d'Adaptation d'Algorithme Génétique .

### 3.4. APPROCHE 2 : DESCRIPTION DE LA STRATÉGIE PROPOSÉE(ACO-DSOS)

De nos jours le transport est un élément essentiel et indissociable de toute société car il relie diverses régions et aide les gens à se déplacer facilement entre différentes destinations.

Le contrôle de la synchronisation des feux de circulation est une question importante dans la conception des systèmes de contrôle de la circulation, car une synchronisation inappropriée pourrait entraîner une forte congestion dans le flux du trafic.

Malheureusement, les chercheurs ne sont pas parvenus à une solution définitive pour résoudre le problème de la congestion du trafic.

Actuellement, la plupart des systèmes de contrôle du trafic utilisent encore des périodes de temps prédéfinies pour contrôler les feux de circulation.

Avec l'augmentation du nombre des automobiles, les feux, et aussi les routes qui contiennent des millions des intersections, l'application de l'algorithme DIJKSTRA devient inefficace car il va nous générer le résultat dans un temps de calcul très long. Les méta-heuristiques sont très utiles pour une telle situation.

L'objectif de la deuxième stratégie est d'appliquer l'algorithme de colonie de fourmis pour trouver les feux critiques qui génèrent un très grand nombre d'automobiles en attente dans un espace de recherche très important qui fera l'objet de la première phase.

La contribution principale de la deuxième phase est d'implémenter l'algorithme Discret symbiotic Organism (DSOS) sur le problème d'optimisation des feux de signalisations.

Nous avons proposé cette méta-heuristique pour une solution qui améliore le trafic et qui aide les conducteurs à se déplacer facilement.

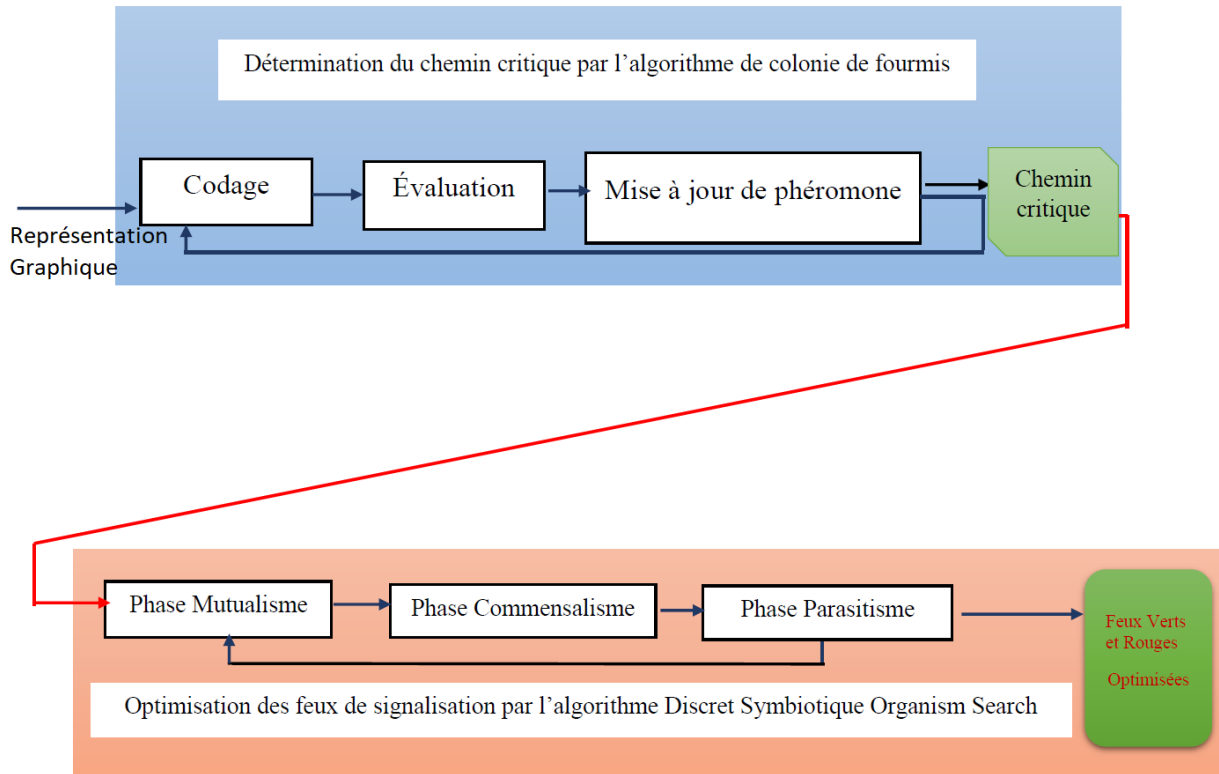


FIGURE 3.7 – Architecture générale de l'approche(ACO-DSOS).

### 3.4.1. NIVEAU 1 : RECHERCHE DES FEUX CRITIQUES PAR L'ALGORITHME DE COLONIE DES FOURMIES (ACO)

Nous présenterons dans notre thèse l'application d'une méta-heuristique permettant de résoudre le problème de détermination des feux de circulation qui engendrent l'attente des véhicules sur les points des feux, en s'inspirant du comportement social des fourmis. Le comportement des fourmis est un comportement collectif.



Chaque fourmi a pour priorité le bien être de la communauté. Chaque individu de la colonie est à priori indépendant et n'est pas supervisé d'une manière ou d'une autre. Ce concept est appelé Hétérarchie (s'opposant à la Hiérarchie), chaque individu est aidé par la communauté dans son évolution et en retour il aide au bon fonctionnement de celle-ci.

En observant une colonie de fourmis à la recherche de nourriture dans les environs du nid, on s'aperçoit qu'elle résoud des problèmes tels que celui de la recherche du plus court chemin. Les fourmis résolvent des problèmes complexes par des mécanismes assez simples à modéliser. Il est ainsi assez simple de simuler leur comportement par des algorithmes.

### **3.4.1.1. ADAPTATION DE COLONIE DE FOURMIS SUR LA DÉTERMINATION DES FEUX DE SIGNALISATION CRITIQUES**

L'algorithme repose sur un ensemble de fourmis, chacune parcourant un chemin parmi ceux possibles. À chaque étape, la fourmi choisit de passer d'un feu à un autre en fonction de quelques règles.

- Une fourmi choisit un chemin, et trace une piste de phéromone.
- L'ensemble des fourmis parcourt un certain nombre de chemins, chaque fourmi déposant une quantité de phéromone proportionnelle à la qualité du parcours.
- La concentration des phéromones élevée dans un chemin indique l'encombrement élevée dans le chemin lui même comme montre la figure 3.8.
- L'évaporation des phéromones fait disparaître les mauvaises solutions (chemin moins encombré).

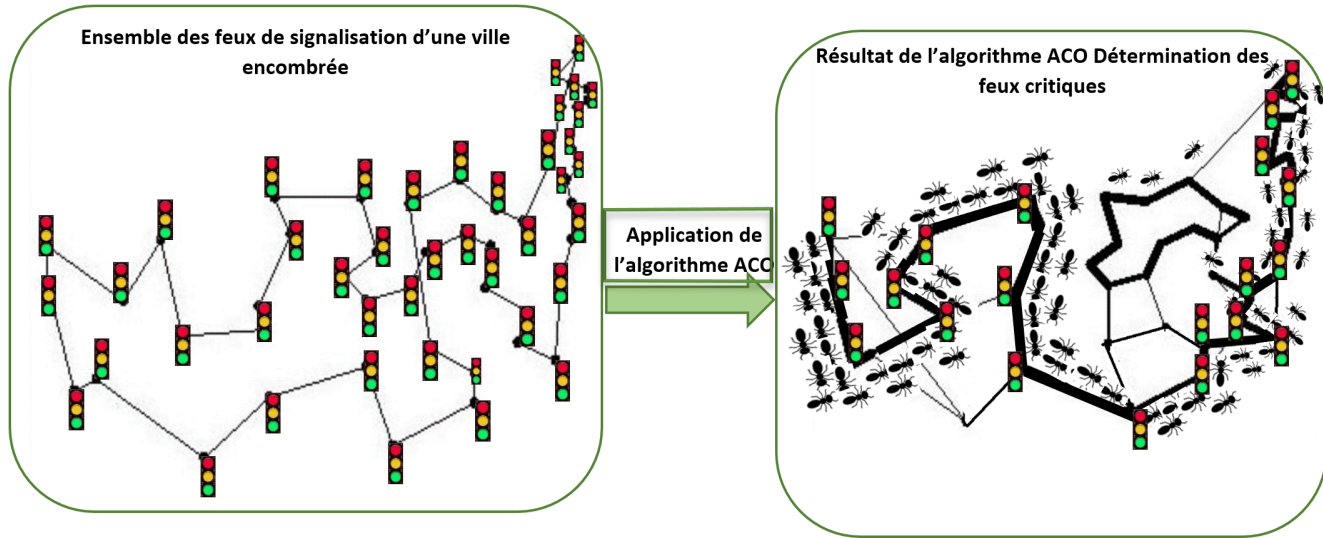


FIGURE 3.8 – Application de l'algorithme Ant Colony.

Les fourmis  $m$  qui ont construit une solution dans la boucle peut mettre à jour le phéromone. La valeur de la phéromone  $\tau_{ij}$  qui en contact avec l'ane  $ij$  entre le feux  $i$  et le feux  $j$  est mis à jour avec la formule suivante :

$$\tau_{ij} = (\rho) * \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k \quad (3.8)$$

Le paramètre  $\rho$  est le taux d'évaporation des phéromones,  $m$  est le nombre de fourmis,  $\Delta\tau_{ij}^k$  est la quantité de phéromone laissée sur le lane  $(i,j)$  par la fourmi  $k$  :

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L^k} & \text{k la fourmi se déplace sur le feux i à j} \\ 0 & \text{otherwise} \end{cases} \quad (3.9)$$

Où  $Q$  est une constante,  $L_k$  est la longueur du tour qui construit par ant  $k$  dans la boucle actuelle. Au cours du processus de construction, les fourmis visiteront les feux de circulation suivante via un mécanisme stochastique : Alors qu'une fourmi se trouvant dans le feu de signalisation  $i$  a construit la solution partielle, la probabilité de se déplacer vers le feu  $j$  est donnée par la formule suivante :

$$P_{ij}^k(t) = \frac{[\tau_{ij}]^\alpha * [\mu_{ij}]^\beta}{\sum_{L \in N_t^k} [\tau_{ij}]^\alpha * [\mu_{ij}]^\beta} \quad (3.10)$$

Le paramètre  $N_t^k$  est ici un ensemble d'éléments convenables. Autrement dit, il s'agit d'un ensemble d'arêtes  $(i,1)$  ici le paramètre 1 désigne le feu non visité par ant  $k$ .

Les paramètres  $\alpha$  et  $\beta$  entrent en contact avec l'importance entre la phéromone et l'information heuristique qui a été donnée par la formule suivante :

$$\mu_{ij} = \frac{1}{d_{ij}} \quad (3.11)$$

Où le paramètre  $d_{ij}$  indique le nombre d'automobiles entre le feu de circulation  $i$  et  $j$ .

### 3.4.2. NIVEAU02 : OPTIMISATION DES FEUX DE SIGNALISATION PAR L'ALGORITHME DISCRET SYMBIOTIC ORGANISM SEARCH

#### 3.4.2.1. SYMBIOTIC ORGANISM SEARCH

Une optimisation efficace des feux de signalisation est l'une des étapes

majeur dans cette thèse, l'algorithme Symbiotic Organism Search proposé simule le comportement interactif observé parmi les organismes dans la nature. Les organismes vivent rarement isolés en raison de la dépendance à l'égard d'autres espèces pour leur subsistance et même leur survie. Cette relation basée sur la dépendance est connue sous le nom de symbiose. La sous-section suivante clarifie la signification de la symbiose, donne des exemples d'archétypes de relations symbiotiques et décrit le rôle de la symbiose dans l'écosystème[109]. cette étape consiste à appliquer une méta-heuristique nommé symbiotic organism search dans l'objective de minimiser le nombre des véhicules en attente.

Les algorithmes métaheuristiques actuels imitent les phénomènes naturels. Par exemple, la colonie d'abeilles artificielles (ABC) simule le comportement de recherche de nourriture des essaims d'abeilles, l'optimisation des essaims de particules simule le comportement de rassemblement des animaux et l'algorithme génétique simule le processus d'évolution naturelle. SOS simule les interactions symbiotiques au sein d'une relation d'organismes appariés qui sont utilisées pour rechercher l'organisme le plus apte. L'algorithme proposé a été développé initialement pour résoudre l'optimisation numérique sur un espace de recherche continu.

SOS commence par une population initiale appelée l'écosystème. Dans l'écosystème initial, un groupe d'organismes est généré aléatoirement dans l'espace de recherche. Chaque organisme représente une solution candidate au problème correspondant. Chaque organisme de l'écosystème est associé à une certaine valeur de fitness, qui reflète le degré d'adaptation à l'objectif souhaité.

La génération de nouvelles solutions est régie par l'imitation de l'interaction biologique entre deux organismes de l'écosystème. Trois phases qui ressemblent au modèle d'interaction biologique du monde réel sont introduites : la phase de mutualisme, la phase de commensalisme et la phase de parasitisme.

Le caractère de l'interaction définit le principe de base de chaque phase. Les interactions profitent aux deux parties dans la phase de mutualisme ; profitent à un côté et n'impactent pas l'autre dans la phase de commensalisme ; profiter d'un côté et nuire activement à l'autre dans la phase de parasitisme. Chaque organisme interagit avec l'autre organisme au hasard à travers toutes les phases. Le processus est répété jusqu'à ce que les critères de terminaison soient satisfaits[109].

### 3.4.2.2. ADAPTATION D'ALGORITHME DISCRET SYMBIOTIC ORGANISM SEARCH SUR LE PROBLÈME DES FEUX DE SIGNALISATION

La recherche d'organismes symbiotiques (SOS) est une technique d'optimisation métaheuristique récemment développée pour résoudre des problèmes d'optimisation numérique. SOS imite les relations symbiotiques (mutualisme, commensalisme et parasitisme) présentées par les organismes dans un écosystème.

Notre objectif est de minimiser le temps d'attente des automobiles en trouvant le meilleur groupe de temps (feux vert, feux rouge) à exécuter sur les intersections. La valeur de fitness de chaque organisme qui détermine la force du niveau d'adaptation de l'organisme à l'écosystème.

#### DESCRIPTION DE L'ORGANISME

SOS démarre son processus en générant d'abord au hasard un nombre  $N$  d'organismes pour peupler l'écosystème. chaque organisme représente dans ce cas une solution candidate au problème des feux (vert,rouge) avec une fonction objectif (3.7)(réduire le nombre des véhicules en attente). Le processus de recherche commence immédiatement après la création de l'écosystème initial, l'écosystème est un ensemble d'organisme et l'organisme est un ensemble des feux (vert, rouge). Les trois phases de symbiose considérées ici comprennent la phase de mutualisme, la phase de commensalisme et la phase de parasitisme.

## Approches Proposées

---

Les étapes de l'algorithme Symbiotic Organism Search sont données ci-dessous :

- **1. Initialisation de l'écosystème** : La population initiale de l'écosystème est générée et d'autres variables de contrôle telles que taille de l'écosystème, le nombre maximal d'itérations est spécifié. Les postes des organismes dans l'espace des solutions sont représentés par des nombres entiers (feux vert et feux rouge).

Soit une fonction  $f : D \rightarrow \mathbb{R}$  trouver  $x' \in D$  (minimisation), Où  $f$  est une fonction objectif à optimiser (la fonction (3.7) ) et  $D$  représente l'espace de recherche tandis que les éléments de  $D$  sont les solutions réalisables.  $X$  est un vecteur d'optimisation.  $X = x_1, x_2, x_3, \dots, x_n$  Une solution optimale est une solution réalisable  $X'$  qui optimiser la fonction (3.7).

l'écosystème est exprimé par  $X = \{x_1, x_2, x_3, \dots, x_n\}$ . La position du  $i$ ème organisme exprimée comme un vecteur du 1 élément.  $X_i$  représente les feux verts et les feux rouges pour un cycle (organisme), peut être donnée comme  $X_i = [x_{i1}, x_{i2}, x_{i3}, x_{i4}]$ . Ainsi, chaque organisme représente une solution potentielle au problème.

- **2. Sélection** de l'organisme avec la fonction objectif (3.7) la mieux adaptée représentée par  $x_{best}$ , meilleur feux vert et feux rouge qui donne un minimum de temps d'attente des véhicules.
- **3. Phase mutualisme** : Dans la  $i$ ème itération, un organisme qui contient un ensemble des feux (vert et rouge)  $X'_{jgreen}, X'_{jred}$  est choisi au hasard dans l'écosystème pour interagir avec un organisme  $X'_{igreen}, X'_{ired}$  pour un bénéfice mutuel avec  $i \neq j$  selon les formules suivantes :

$$X'_{igreen} = |X_{igreen} + r'(x_{best} - (\frac{X_{igreen} + X_{jgreen}}{2}) * F_1)| \quad (3.12)$$

$$X'_{red} = |X_{red} + r'(x_{best} - (\frac{X_{red} + X_{red}}{2}) * F_1)| \quad (3.13)$$

$$X'_{green} = |X_{green} + r''(x_{best} - (\frac{X_{green} + X_{green}}{2}) * F_2)| \quad (3.14)$$

$$X'_{red} = |X_{red} + r''(x_{best} - (\frac{X_{red} + X_{red}}{2}) * F_2)| \quad (3.15)$$

où  $r'$  et  $r''$  sont des nombres aléatoires générés uniformément entre 0 et 1.  $f_1$  et  $f_2$  sont les facteurs de bénéfice mutuel résultant de l'association entre les organismes de la relation. Les deux organismes peuvent bénéficier également de la relation mutuelle ou l'un peut bénéficier plus que l'autre. Les valeurs de  $f_1$  et  $f_2$  sont déterminées au hasard. la première indique un bénéfice partiel tandis que la deuxième indique un bénéfice complet.

Le terme  $(\frac{x_i+x_j}{2})$  définit le degré d'adaptation des deux espèces à l'écosystème.

nous avons calculé les nouveaux feux vert et rouge ( $X'_{green}, X'_{red}, X'_{green}, X'_{red}$ ) avec les équations (3.12), (3.13), (3.14), (3.15) ci dessus dans la phase mutualisme et en appliquant la fonction objective (3.7) et tester avec la fonction objective précédente et en gardant la meilleur fonction objective qui contient les meilleurs feux avec un minimum du temps d'attente.

- **4. Phase commensalisme** : Dans la ième itération, un organisme  $X_{green}, X_{red}$  est choisi au hasard dans la population pour interagir avec  $X_{green}, X_{red}$  où  $i \neq j$ . Dans cette phase,  $X_{green}, X_{red}$  profite de  $X_{green}, X_{red}$  mais  $X_{green}, X_{red}$  ne gagne ni ne perd d'interaction. L'interaction est modélisée selon les formules suivantes :

$$X'_{green} = |X_{green} + r'(x_{best} - X_{green})| \quad (3.16)$$

$$X'_{red} = |X_{red} + r'(x_{best} - X_{red})| \quad (3.17)$$

où  $r'$  est un nombre aléatoire généré uniformément entre 0 et 1, nous avons recalculé le nouveau feu de l'organisme  $i$  ( $X_{green}, X_{red}$ ) et en appliquant la fonction objective (3.7) et en teste avec la précédente et en garde la meilleur séquence des feux.

- **5. Phase Parasitisme** : Dans la  $i$ ème itération, un vecteur parasite ( $X_{p_{green}}, X_{p_{red}}$ ) est créé en mutant ( $X_{green}, X_{red}$ ) en utilisant un nombre généré aléatoirement dans la plage des variables sélectionné au hasard dans la population pour servir d'hôte à ( $X_{p_{green}}, X_{p_{red}}$ ). Si la valeur de la fonction objective d'organisme  $X_{green}, X_{red}$  est supérieure de vecteur parasite, alors  $X_{p_{green}}, X_{p_{red}}$  remplacera  $X_{green}, X_{red}$ , sinon  $X_{p_{green}}, X_{p_{red}}$  est ignoré.

$$X_{p_{green}} = |X_{green}| \quad (3.18)$$

$$X_{p_{red}} = |X_{red}| \quad (3.19)$$

- Les étapes 2 à 5 sont répétées jusqu'à ce que le critère d'arrêt soit atteint.

cette étape consiste à appliquer une méta-heuristique nommé Symbiotic Organism Search Algorithm dans l'objective de minimiser le nombre de véhicule en attente.

l'organigramme ci dessus présente l'adaptation détaillé du symbiotic organism search sur le problème des feux de signalisation.



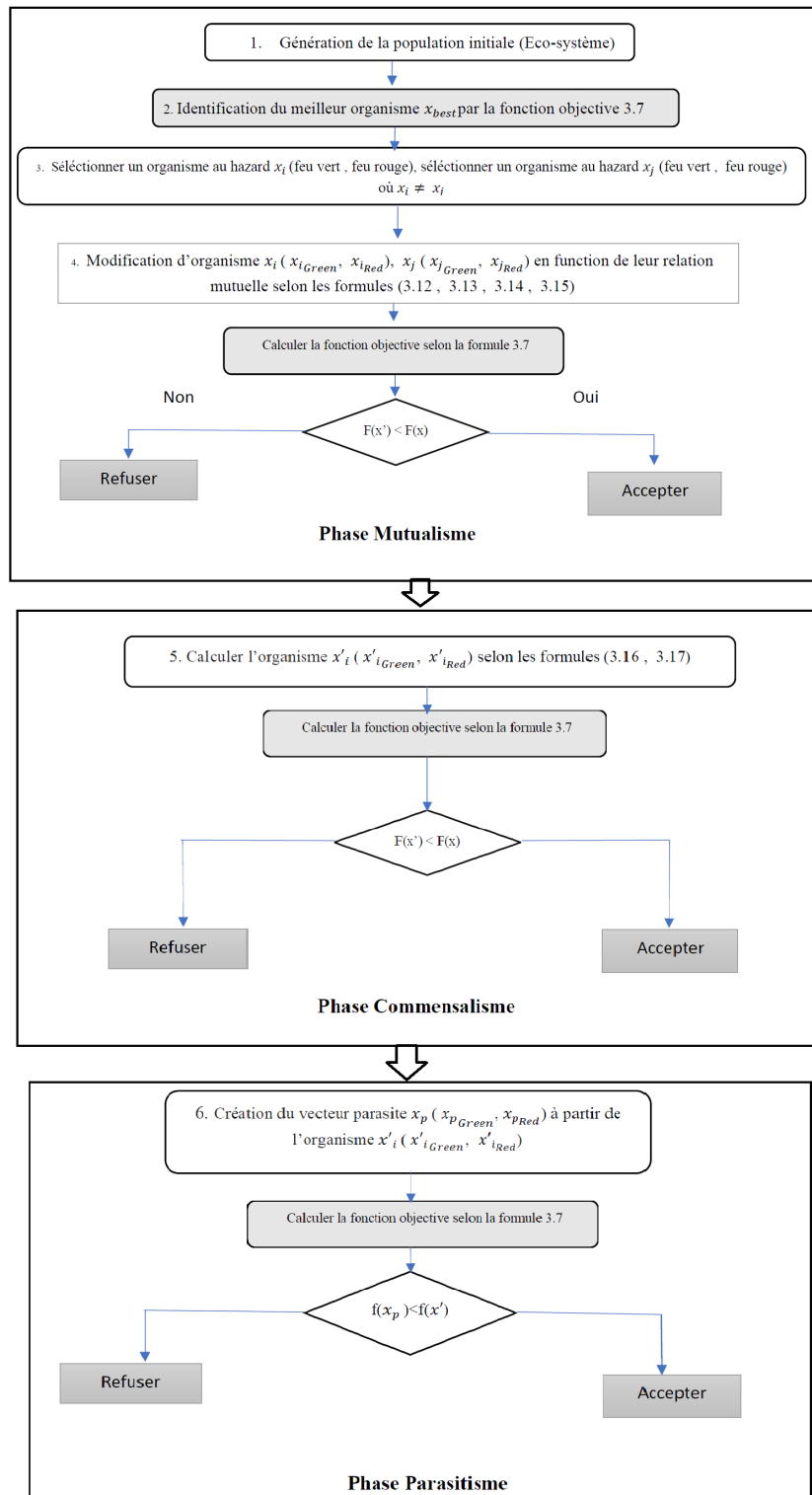


FIGURE 3.9 – Organigramme d'Adaptation d'algorithmme symbiotic organism search.

### 3.5. CONCLUSION

Au cours de ce chapitre, nous avons décrit nos approches sur l'optimisation des feux de signalisation. Nous avons présenté les différentes méthodes. Nous avons aussi étendu nos approches par l'ajout des méthodes inspirées de la nature nommée Génétique algorithm (GA), symbiotic organism search (SOS).

Dans le chapitre suivant, nous nous intéressons à la concrétisation de nos stratégies proposées et cela par l'implémentation et l'interprétation des différents résultats obtenus.

# Chapitre 4

## RÉSULTATS EXPERIMENTAUX

### Sommaire

---

4.1	Introduction	91
4.2	Langage et Environnement de travail	91
4.2.1	Langage de programmation JAVA	91
4.2.2	Environnement de travail Netbeans	91
4.2.3	Python	92
4.2.4	Simulateur SUMO	92
4.3	Étude de cas : régulation du trafic de la zone d'Alger (Bir morad rais - El moradia - Hydra)	95
4.4	Résultats Expérimentaux	98
4.4.1	Résultats de l'approche d'optimisation de feux basée sur les Algorithmes Génétiques	98
4.4.2	Résultats de l'approche d'optimisation de feux basée sur l'Algorithme Discret Symbiotique Organisme Search (DSOS)	104
4.5	Conclusion	108
	Bibliographie	111

---

## 4.1. INTRODUCTION

Ce chapitre est consacré à la phase d'implémentation de nos stratégies d'optimisation des feux de signalisation. Il permettra d'évaluer et de valider nos stratégies proposées sur plusieurs intersections. Nous avons réalisé plusieurs simulations, dans le but d'effectuer des séries d'expérimentations dont les résultats font l'objet de ce chapitre.

## 4.2. LANGAGE ET ENVIRONNEMENT DE TRAVAIL

### 4.2.1. LANGAGE DE PROGRAMMATION JAVA

Le langage Java est un langage de programmation orienté objet. Java a la particularité principale d'être portable. Nous nous sommes orientés sur le langage JAVA qui présentait néanmoins beaucoup d'avantages. Ce langage est apparu vers la fin de 1995 et obtient rapidement un énorme succès. Il s'agit d'un langage de conception très performant qui a été adopté par la majorité des scientifiques. Ses caractéristiques intégrées de sécurité offrent un sentiment de confiance aux programmeurs comme aux utilisateurs des applications. De plus, Java incorpore des fonctionnalités qui facilitent grandement certaines tâches de programmation avancées comme la gestion des réseaux, la connectivité des bases de données ou le développement d'applications multitâches, etc [110].

### 4.2.2. ENVIRONNEMENT DE TRAVAIL NETBEANS

Netbeans a été créé à l'initiative de Sun Microsystems. Il présente toutes les caractéristiques indispensables à un environnement de qualité, que ce soit pour développer en Java, Ruby, C/C++ ou même PHP. Cet environnement de développement (IDE) est sous licence Open Source, il permet de développer et déployer rapidement et gratuitement des applications graphiques Swing, des Applets, des JSP/Servlets, des architectures J2EE, dans un environnement fortement personnalisable. En se reposant sur un noyau robuste, la plateforme NetBeans, que vous pouvez également utiliser pour développer vos propres applications Java, est un système de plugins performant, qui permet d'avoir un

IDE modulable [110]. Toutes les expériences ont été exécutées sur un système avec un processeur Intel Core i7 avec 1,8 GHz, 8 Go de mémoire et un système d'exploitation Windows 10 64 bits.

#### 4.2.3. PYTHON

Le langage de programmation Python a été créé en 1989 par Guido van Rossum, aux Pays-Bas. La première version publique de ce langage a été publiée en 1991. Python est un langage de programmation interprété multi-paradigme. Il favorise la programmation impérative structurée, et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions [112].

#### 4.2.4. SIMULATEUR SUMO

"Simulation of Urban MObility" (SUMO) est un progiciel de simulation de trafic open source, hautement portable, microscopique et continu, conçu pour gérer de grands réseaux. Il permet une simulation intermodale incluant les piétons et est livré avec un large ensemble d'outils pour la création de scénarios. Il est principalement développé par des employés de l'institut des systèmes de transport du centre aérospatial allemand [113].



FIGURE 4.1 – Simulation of Urban MObility (SUMO).

#### 4.2.4.1. PRINCIPE DE FONCTIONNEMENT

SUMO permet de modéliser le réseau routier urbain en définissant les routes, les intersections, les voies de circulation, les feux de signalisation, les limitations de vitesse, etc (figure 4.2). Ces éléments sont configurés à l'aide de fichiers au format XML. Il génère des flux de véhicules qui suivent des itinéraires prédéfinis. Les véhicules dans SUMO sont dotés de comportements réalistes, tels que l'accélération, le freinage, le changement de voie, etc. Ces comportements sont modélisés à l'aide d'algorithmes basés sur des règles et des modèles de comportement. Il calcule les interactions entre les différents véhicules sur le réseau routier, ainsi qu'avec les infrastructures telles que les feux de signalisation et les panneaux de signalisation. Cela permet de simuler des situations de trafic complexes et réalistes. Une fois la simulation terminée, SUMO fournit une gamme de résultats et de statistiques, tels que les temps de parcours, les émissions de polluants, les performances du réseau routier, etc [114].

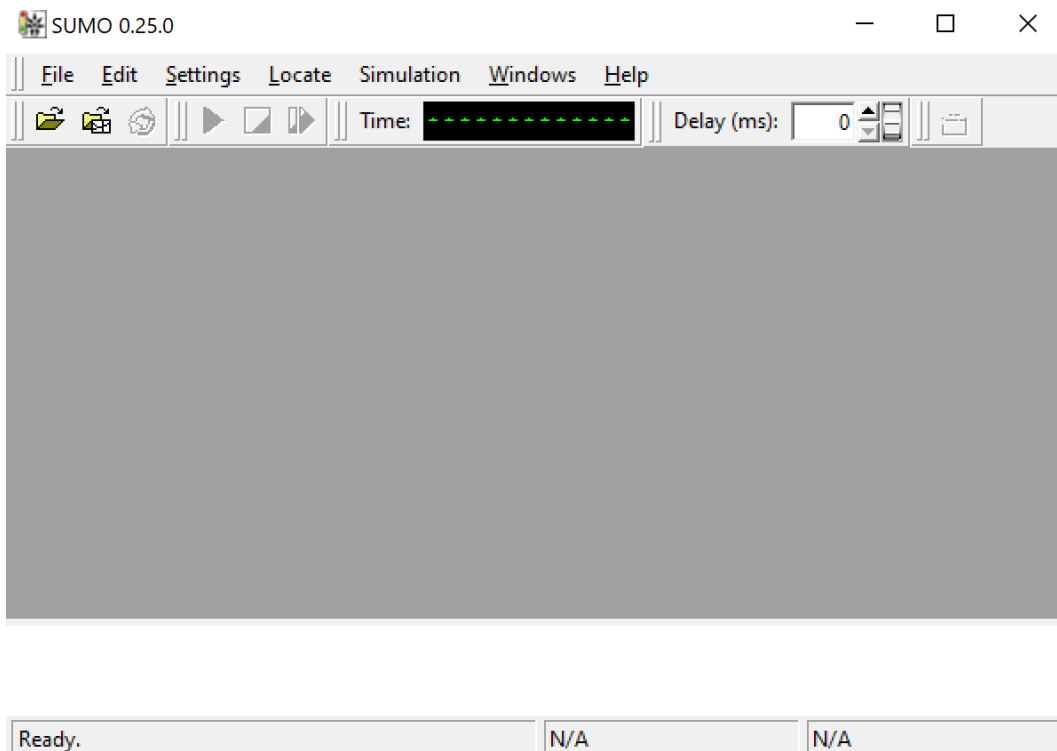


FIGURE 4.2 – Interface graphique du SUMO.

#### 4.2.4.1.1 Applications incluses :

Le paquet SUMO est impliqué dans plusieurs projets de recherche nationaux et internationaux. Les applications incluses couvrent une gamme de domaines tels que l'évaluation des feux de signalisation, le choix et le re-routage d'itinéraires, l'évaluation des méthodes de surveillance du trafic, la simulation des communications véhiculaires et les prévisions de trafic.

SUMO est un logiciel complet qui comprend plusieurs applications nécessaires à la préparation de la simulation. La simulation de trafic avec SUMO exige une représentation précise des réseaux routiers et de la demande de trafic, qui peuvent être obtenus de différentes manières. Les réseaux routiers SUMO peuvent être créés grâce à l'outil "netgenerate" ou importés depuis une carte de route numérique.

Dans le paquet SUMO, plusieurs applications principales sont incluses pour aider à la modélisation, à la simulation et à l'analyse de la mobilité urbaine.

Voici des principales applications incluses :

TABLE 4.1 – Paquet SUMO.

Nom de l'application	Description
SUMO	La simulation microscopique sans interface graphique est exécutée via une application en ligne de commande.
SUMO-CONSOLE	La simulation microscopique mode console.
SUMO-GUI	La simulation microscopique avec une interface utilisateur graphique.
NETCONVERT	Outil d'importation et de génération de réseau, il permet de lire les configurations routières provenant de divers formats et les convertit au format utilisé par SUMO.
NETEDIT	Un éditeur de réseau graphique.
NETGENERATE	Permet de générer en ligne de commande un réseau de plusieurs formes : en grille, en toile ou aléatoire.

### 4.3. ÉTUDE DE CAS : RÉGULATION DU TRAFIC DE LA ZONE D'ALGER (BIR MORAD RAIS - EL MORADIA - HYDRA)

L'une des prémisses pour pouvoir simuler des activités de déplacement dans un réseau routier donné est de répondre à la demande de trafic.

Nous considérons un scénario dérivé du trafic réseau d'Alger, la zone urbaine sélectionnée est délimitée au Nord El Biar, à l'Ouest par la rue Val d'Hydra, à l'Est par El Madania, et au Sud par Saïd Hamdine. Lequel couvre une superficie d'environ 1 km<sup>2</sup> avec 21 voies contrôlées par des feux de signalisation. La figure 4.3 montre une capture de la zone étudiée.

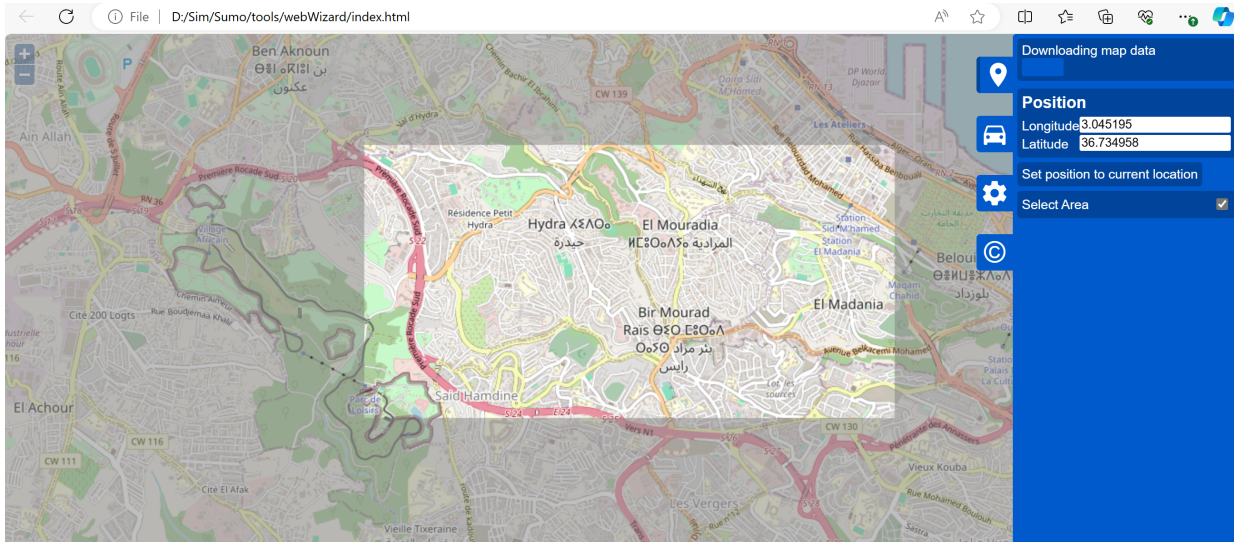


FIGURE 4.3 – Sélection de la zone étudiée.

Du moment où la zone est sélectionnée, on clique sur le bouton "Generate Scenario" pour générer le scénario. OsmWebWizard initie le processus en récupérant les données OpenStreetMap spécifiques à la zone géographique sélectionnée. Ces données sont riches en informations détaillées sur les routes, les intersections, les feux de signalisations, les voies ...etc.



Les données du réseau sont extraites via **osmWebWizard** puis transformées en un format compatible avec SUMO (osm.net.xml) à l'aide de scripts fournis dans le package SUMO. Une fois que le scénario est généré avec **osmWebWizard**, cette plateforme ouvre automatiquement l'interface graphique SUMO-GUI, préconfigurée avec le fichier de configuration osm.sumocfg. Cela signifie que dès que vous avez terminé de définir votre scénario de simulation de trafic dans **osmWebWizard**, vous pouvez immédiatement visualiser et exécuter ce scénario dans SUMO-GUI (figure 4.4).

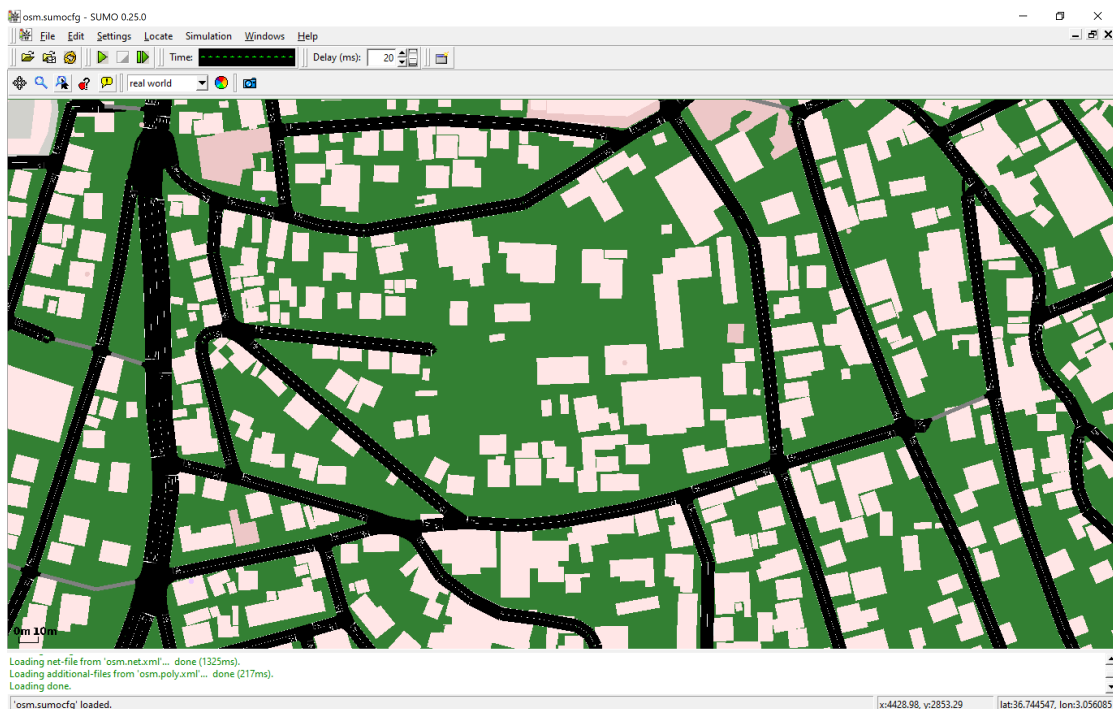


FIGURE 4.4 – Visualisation du scénario créé par osmWebWizard .

Un scénario de trafic est créé, définissant les véhicules, leurs itinéraires, leur comportement, ainsi que les conditions initiales telles que la densité de trafic et les événements spéciaux. Une fois la simulation terminée, SUMO fournit des données détaillées sur divers aspects du trafic, tels que les temps de trajet, les flux de véhicules, les congestions, etc. Cette section expose des détails comme indiqué à la figure 4.5.

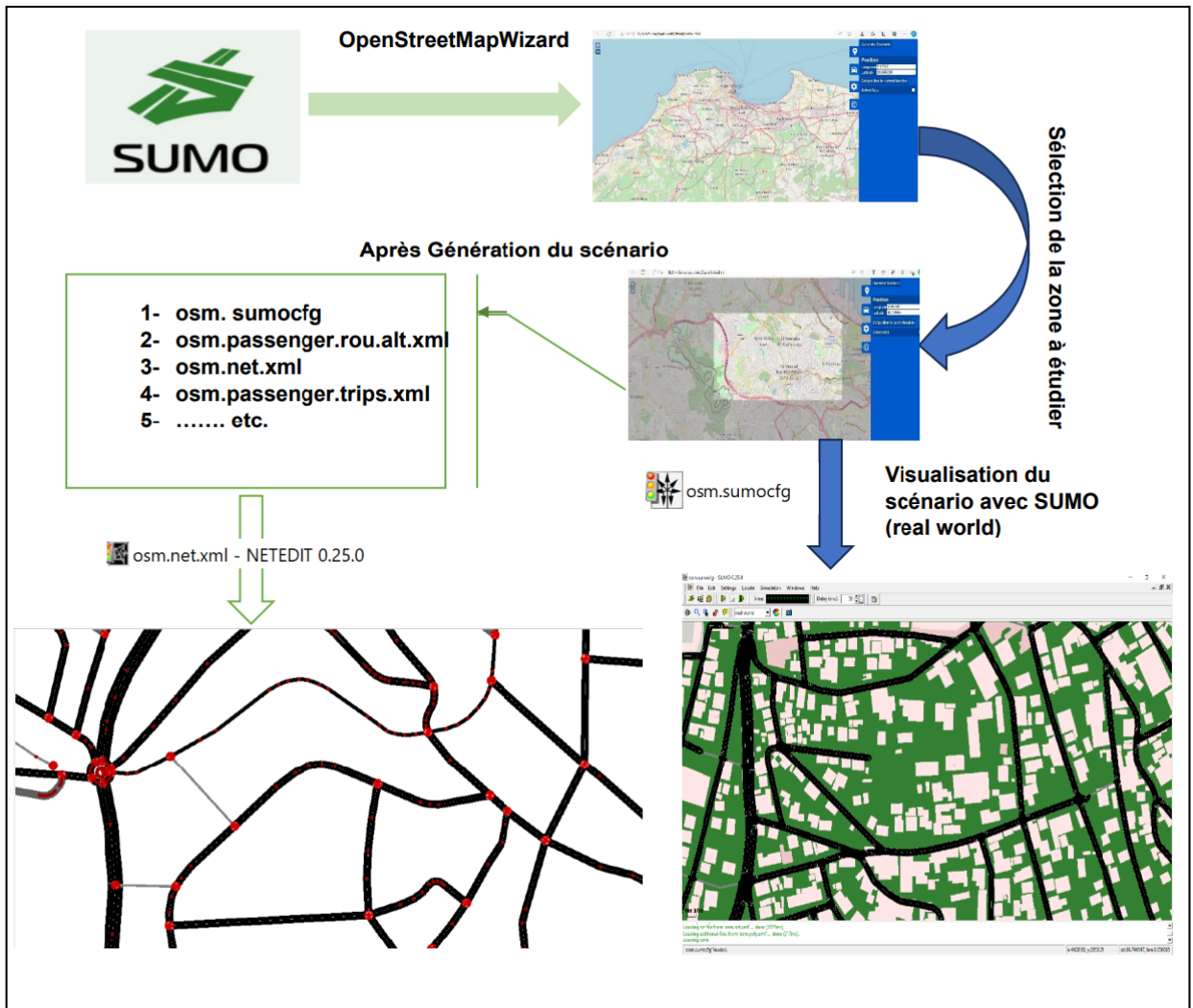


FIGURE 4.5 – Structure globale pour le traitement de la zone étudiée.

L'objectif fondamental de cette simulation est de mener une étude approfondie de notre modèle de trafic. Cela implique de le soumettre à une série de tests dans des conditions proches de la réalité, afin d'évaluer sa performance et sa fiabilité. En examinant attentivement les résultats de la simulation, nous visons à obtenir une compréhension détaillée du comportement du trafic dans le réseau modélisé. ndie nous permettra de décrire de manière exhaustive les différentes interactions et dynamiques observées, offrant ainsi des informations précieuses pour affiner et améliorer notre modèle.

## 4.4. RÉSULTATS EXPÉRIMENTAUX

### 4.4.1. RÉSULTATS DE L'APPROCHE D'OPTIMISATION DE FEUX BASÉE SUR LES ALGORITHMES GÉNÉTIQUES

#### 4.4.1.1. CONFIGURATION DE SIMULATION

Afin de mieux démontrer l'impact du cadre développé, la première chose à effectuer est de déterminer le nombre des feux à gérer. Le simulateur du trafic possède des entrées qui sont utilisées par ses différents modules de son corps (les feux vert et feux rouge, le nombre des véhicules arrivées en feux vert et feux rouge, le nombre des véhicules libérées, la voie, le cycle qui contient plusieurs phase). Alors que le cas étudie un plan de signalisation qui inclus plusieurs intersection.

Le simulateur que nous avons développé se compose de plusieurs parties principales, qui incluent toutes les fonctions de notre simulation. La première partie comporte un champs pour préciser le le nombre d'intersections obtenue à partir des fichiers générés par SUMO pour la zone étudiée et un bouton **simulation** qui applique nos algorithmes sur nos données réels. Du moment où le nombre des feux est spécifié, on peut passer au choix d'algorithme de prétraitement utilisés (DIJKSTRA). Nous avons appliqué l'algorithme DIJKSTRA qui permet de déterminer le chemin critique au lieu de traiter toute les liens possibles.

La troisième partie est la partie la plus importante qui permet d'appliquer l'algorithme génétique afin d'optimiser le nombre des véhicules en attente. Il comprend deux champs : la taille de la population et le nombre d'itération.

Après l'exécution, nous avons comme résultat un chromosome avec les meilleurs feux donnant de meilleur valeur de la fonction objective. L'objectif est d'optimiser les feux de circulation de sorte que la congestion du trafic, le cas échéant, puisse être réduite.

**4.4.1.2. RÉSULTAT 1 : COMPARAISON DE L'APPROCHE STATIQUE AVEC L'APPROCHE DJ-GA**

Dans cette expérience, nous avons mis en œuvre une fonction objective spécifique (3.7)) qui est conçue pour calculer le nombre de véhicules en attente à un moment donné. Cette fonction joue un rôle crucial dans notre simulation car elle nous permet d'évaluer l'efficacité de différentes stratégies de gestion du trafic. L'expérience s'appuie sur les résultats obtenus lors de l'étape de prétraitement, qui identifie l'ensemble des voies de circulation les plus encombrées. Ces voies, que nous appelons "voies critiques", sont les voies 0-9, 9-3, et 3-7. Ces voies ont été identifiées comme étant particulièrement sujettes à la congestion, ce qui en fait des cibles clés pour l'amélioration. Pour cette simulation, nous avons fixé le nombre de population à 1000 et le nombre d'itérations à 1000 (comme indiqué dans le tableau (4.2)). Ces paramètres ont été choisis pour permettre à l'algorithme d'explorer suffisamment l'espace des solutions possibles tout en restant dans des limites de calcul raisonnables. Les résultats de notre simulation montrent une amélioration significative en termes de nombre de véhicules en attente sur plusieurs cycles et plusieurs voies par rapport à une approche statique (voir tableau (4.3)). Cela suggère que notre algorithme GA proposé fonctionne correctement et est capable d'améliorer la gestion du trafic. Après avoir appliqué l'algorithme génétique à notre problème d'optimisation des feux de circulation, nous avons observé une amélioration significative du véhicules en attente. En particulier, le nombre de véhicules en attente moyen a été réduit par rapport à la méthode de contrôle fixe. En comparant les résultats avec ceux obtenus par la méthode statique, nous avons constaté que l'algorithme génétique a surpassé la méthode à feux fixe en termes de réduction du nombre de véhicules en attente. Cela suggère que l'algorithme génétique est particulièrement efficace pour ce type de problème d'optimisation. En ce qui concerne les paramètres de l'algorithme génétique, nous avons constaté que la taille de la population (1000 individus) et les taux de mutation et de croisement ont tous eu un impact significatif sur les performances de l'algorithme. Par exemple, une augmentation de la taille de la population a conduit à une meilleure exploration de l'espace des solutions, mais a également augmenté le temps de calcul.

## Chapitre 4. Résultats Experimentaux

TABLE 4.2 – Résultat de la synchronisation du signal sur les AG.

Lanes (Src-Dst)	Phases	Feux Verts	Feux Rouges	Arrivées des véhicules (feux verts)	Arrivées des véhicules (feux rouges)	Véhicules libérées	Taux des véhicules en attente (veh\s)
0 9	1	13	39	62	29	21	67
	2	18	17	71	48	23	
	3	18	18	78	54	23	
	4	57	69	47	20	20	
0 9	1	59	55	40	40	25	120
	2	15	27	34	60	24	
	3	10	68	54	43	22	
	4	67	22	23	66	25	
0 9	1	40	65	54	41	21	202
	2	13	32	69	50	22	
	3	59	22	51	56	20	
	4	10	25	66	52	21	
0 9	1	16	10	71	75	24	265
	2	36	60	49	22	25	
	3	11	34	73	37	27	
	4	40	57	50	35	26	
9 3	1	16	60	67	22	29	72
	2	14	45	75	77	24	
	3	64	14	75	56	20	
	4	12	25	31	65	26	
9 3	1	60	32	20	61	24	139
	2	15	22	59	69	20	
	3	61	50	33	77	22	
	4	33	58	53	44	20	
9 3	1	10	26	69	53	23	201
	2	64	26	26	65	25	
	3	47	13	58	30	27	
	4	16	26	76	50	24	
9 3	1	15	26	56	62	25	261
	2	43	12	23	69	27	
	3	69	56	43	32	28	
	4	12	25	71	52	28	
3 7	1	24	45	50	25	27	54
	2	34	26	63	23	27	
	3	58	34	42	54	24	
	4	33	16	42	78	21	
3 7	1	41	17	29	77	29	110
	2	63	26	36	57	26	
	3	61	68	59	73	22	
	4	51	33	24	36	26	
3 7	1	42	10	59	71	21	185
	2	13	51	65	26	22	
	3	66	50	68	60	24	
	4	32	68	65	27	22	
3 7	1	42	14	41	66	24	238
	2	58	65	20	32	22	
	3	19	58	44	41	28	
	4	11	13	64	70	20	

## Chapitre 4. Résultats Experimentaux

TABLE 4.3 – Résultat de la synchronisation du signal de trafic sur temps fixe.

Lanes (Src-Dst)	Phases	Feux Verts	Feux Rouges	Arrivées des véhicules (feu vert)	Arrivées des véhicules (feu rouge)	Véhicules libérées	Taux des véhicules en attente (veh\s)
0 9	1	41	41	62	29	21	78
	2	17	59	71	48	23	
	3	34	31	78	54	23	
	4	41	48	47	20	20	
0 9	1	41	59	40	40	25	143
	2	28	22	34	60	24	
	3	45	55	54	43	22	
	4	36	46	23	66	25	
0 9	1	47	12	54	41	21	235
	2	65	18	69	50	22	
	3	33	48	51	56	20	
	4	49	11	66	52	21	
0 9	1	52	65	71	75	24	327
	2	40	11	49	22	25	
	3	59	23	73	37	27	
	4	53	22	50	35	26	
9 3	1	53	19	67	22	29	95
	2	39	50	75	77	24	
	3	56	69	75	56	20	
	4	61	31	31	65	26	
9 3	1	26	49	20	61	24	178
	2	17	57	59	69	20	
	3	27	49	33	77	22	
	4	39	46	53	44	20	
9 3	1	13	37	69	53	23	267
	2	16	52	26	65	25	
	3	47	18	58	30	27	
	4	35	20	76	50	24	
9 3	1	33	27	56	62	25	331
	2	60	36	23	69	27	
	3	54	64	43	32	28	
	4	16	54	71	52	28	
3 7	1	14	23	50	25	27	77
	2	69	31	63	23	27	
	3	63	33	42	54	24	
	4	63	55	42	78	21	
3 7	1	10	53	29	77	29	154
	2	20	41	36	57	26	
	3	46	51	59	73	22	
	4	68	26	24	36	26	
3 7	1	62	35	59	71	21	243
	2	54	38	65	26	22	
	3	32	32	68	60	24	
	4	68	20	65	27	22	
3 7	1	38	35	41	66	24	314
	2	18	66	20	32	22	
	3	51	31	44	41	28	
	4	59	23	64	70	20	

4.4.1.3. IMPACT DE VARIATION DU NOMBRE D'ITÉRATIONS

Dans cette expérience, notre objectif est de voir comment l'augmentation du nombre d'itération peut affecter le nombre de véhicule en attente (Figure (4.6)). Nous avons varié le nombre d'itération par pas de 200 et nous avons fixé le nombre de population à 1000. Dans cette expérience, nous avons utilisé le résultat de l'étape précédente(étape de prétraitement). Le résultat obtenu après l'exécution de l'algorithme DIJKSTRA est le suivant : Lane 0- Lane 9-Lane3 - Lane 7. Dans un premier temps, nous notons un fort avantage de la stratégie GA par rapport a la méthode statique (temps fixe), quand le nombre d'itération augmente, on remarque que GA donne de meilleures performances. Cet avantage devient de plus en plus clair avec l'augmentation du nombre d'itération. Nous pouvons expliquer ce résultat par la façon du choix des chromosomes. Dans GA, nous classons les chromosomes en fonction de leur fitness. Ensuite, l'opérateur de remplacement choisi est l'origine de cette observation. Cela minimise le nombre de véhicule en attente, et aussi, un nombre d'itérations plus élevé donne à l'algorithme plus de temps pour converger vers une solution optimale. Cela est particulièrement utile si l'espace de recherche est grand.

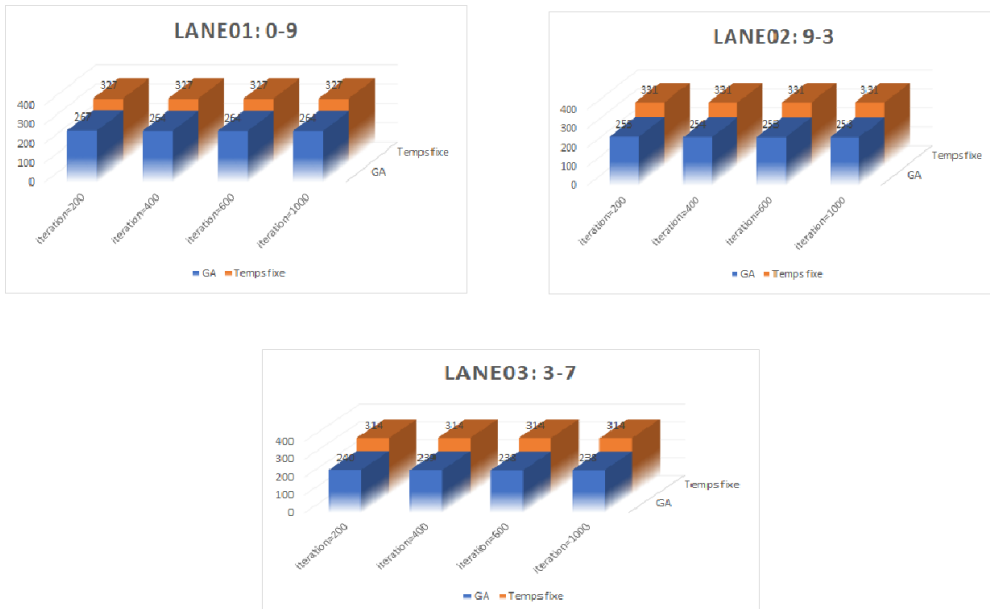


FIGURE 4.6 – Comparaison des Résultat de la synchronisation GA avec Temps fixe.

4.4.1.4. IMPACT DE VARIATION DU NOMBRE DE POPULATION

Pour étudier l'impact du nombre de population sur le nombre de véhicule en attente, nous avons varié le nombre de population de 200 à 1000 et nous avons mesuré le nombre de véhicule en attente comme montre la Figure (4.7). Le nombre d'itération a été fixé à 1000. Les expériences ont été effectués sur les mêmes résultats de l'étape de prétraitement(Lane 0- Lane 9-Lane3 - Lane 7). Nous remarquons une minimisation significative du nombre de véhicule en attente expérimentées avec l'augmentation du nombre de population. Cela est justifié par l'existence d'opérateur de mutation qui permet de provoquer le processus de tenter vers l'optimum. Par conséquent, le nombre de véhicule en attente est réduit. Les résultats de simulations montrent que la stratégie proposée est meilleur par rapport à l'approche statique. Une taille de population plus grande permet une exploration plus large de l'espace des solutions. Cela signifie que l'algorithme a une meilleure chance de trouver l'optimum global.



FIGURE 4.7 – Comparaison des Résultat de la synchronisation du GA avec Temps fixe.



#### **4.4.2. RÉSULTATS DE L'APPROCHE D'OPTIMISATION DE FEUX BASÉE SUR L'ALGORITHME DISCRET SYMBIOTIQUE ORGANISME SEARCH (DSOS)**

##### **4.4.2.1. SIMULATION ET RÉSULTATS**

Dans ce scénario de simulation, nous avons utilisé deux niveaux. Le premier niveau permet d'utiliser l'algorithme de colonie de fourmis pour déterminer les lanes critiques, le deuxième niveau utilise l'algorithme Discret Symbiotique Organisme Search afin de minimiser le nombre de véhicule en attente seulement pour les feux critiques. Les résultats des différentes expérimentations de notre stratégie sont abordés dans cette section.

##### **4.4.2.2. NIVEAU01 : DÉTERMINATION DES FEUX CRITIQUES**

La première étape permet de déterminer le nombre des feux à gérer. Du moment où le nombre des feux sont spécifiés, on peut passer au choix d'algorithme (colonie des fourmis).

L'algorithme de colonie de fourmis est inspiré du comportement des fourmis réelles et constitue une famille de méta-heuristique d'optimisation. En effet les fourmis sont capables collectivement de trouver les lanes critiques.

L'objectif de l'utilisation de cette méta-heuristique est de faire face aux situations où le nombre de feux est très important ce qui pose le problème de la détermination des lanes critiques dans un temps très lent.

Nous avons appliqué cet algorithme aux réseaux d'intersection embouteillés afin de trouver les lanes encombrés.

La sortie de cette étape est l'entrée du deuxième niveau (Lane 0- Lane 9-Lane 3 - Lane 7).

##### **4.4.2.3. NIVEAU02 : APPLICATION D'ALGORITHME DISCRET ORGANISM SEARCH (DSOS)**

Le deuxième niveau est la partie la plus importante dans la simulation, l'approche proposé optimise les feux tricolores en prenant en compte les performances du réseau.

Cette partie présente les résultats obtenus de l'approche Discret Symbiotique Organism Search sur le problème des feux de signalisation.

L'algorithme de recherche d'organismes symbiotiques est un algorithme méta-heuristique récent très prometteur. il a été introduit en considérant les relations entre les créatures dans un écosystème naturel.

Le SOS original utilise trois stratégies d'interaction : le mutualisme, le commensalisme et le parasitisme, pour améliorer une solution candidate.

Nous avons appliqué l'algorithme DSOS sur plusieurs lanes, l'algorithme DSOS est proposé et testé sur un carrefour avec différentes paramètres (le nombre des véhicules arrivées en feux vert et en feux rouge, les feux vert et feux rouge, le nombre des véhicules libérées , la voie, le cycle qui contient plusieurs phase ). Dans cette expérience nous avons utilisé la fonction objective (3.7) qui permet de calculer le nombre de vehicules en attente.

Cette expérience utilise le résultat obtenu dans l'étape précédente qui représente l'ensemble des lanes :(lanes critiques : Lane 0,Lane 9, Lane 3,Lane 7), Dans cette simulation, nous avons fixé le nombre de population à 1000 ainsi que le nombre d'itérartion à 1000 (voir tableau (4.2)).

Nous remarquons qu'il y a une amélioration significative en terme du nombre de véhicule en attente dans plusieurs cycles et plusieurs lanes par rapport à l'approche statique (voir tableau (4.3)). Les résultats de cette expérience montrent que l'algorithme DSOS proposée fonctionne d'une manière correcte. Nous pouvons expliquer cette amelioration par l'augmentation du nombre d'itérations à 1000 avec 1000 population.

### **résultat 01 : Étude Comparative entre l'algorithme DSOS et l'approche statique**

Cette section présente les résultats de simulation des expériences, et la comparaison avec l'approche statique. Nous nous sommes d'abord intéressés à analyser le nombre des véhicules en attente en se basant sur l'algorithme DSOS.

Notre objectif est de mesurer l'impact de l'algorithme DSOS sur le problème des feux de signalisation. Nous avons utilisé le résultat de l'étape précédente(Niveau01).

Le résultat obtenu après l'exécution de l'algorithme ACO est le suivant : Lane 0-

## Chapitre 4. Résultats Experimentaux

Lane 9-Lane 3-Lane 7.

Nous avons fixé la taille de l'éco-système à 1000 ainsi que le nombre d'iteration à 1000. Le scénario de la ville est divisée en 4 cycles, Chaque cycle contient 4 phases, chaque phase représente un interval de temps d'une journée.

Nous remarquons d'après les résultats de la figure (4.8) que la qualité des solutions obtenues par DSOS est meilleure que celle de l'approche statique surtout lorsque le nombre d'iteration est plus élevé.

Nous notons aussi qu'il y a une augmentation du nombre des vehicules en attente d'un cycle à un autre, nous pouvons expliquer cette augmentation par l'ajout du nombre des vehicules en attente du cycle précédent. Comme on peut l'observer, l'approche DSOS a démontré la capacité d'améliorer sa qualité de solution aux stades du processus de recherche.

On pense que l'amélioration des performances par DSOS est attribuée aux trois opérateurs : Mutualisme, Commensalisme, Parasitisme.

Ces opérateurs ont une probabilité plus élevée d'obtenir des solution progressive. DSOS est capable de converger vers une solution presque optimale.

Les résultats comparatifs sont présentés dans les graphes suivants.

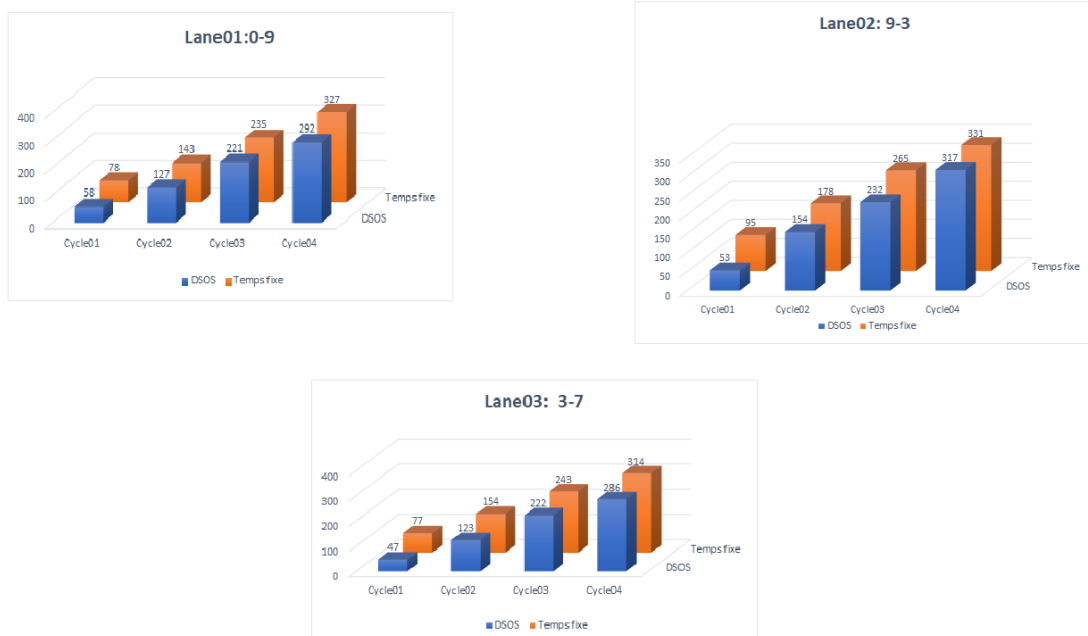


FIGURE 4.8 – Comparaison des Résultat de la synchronisation du DSOS avec Temps fixe.

**résultat 02 : Impact des 3 Operateurs sur le nombre de vehicule en attente**

Pour évaluer les performances de l'algorithme Discret Symbiotic Organism Search, deux combinaisons différentes : la taille de l'éco-système et les iterations ont été fixés.

Les résultats indiquent la moyenne des centaines de simulation, l'effet de l'approche DSOS sur la minimisation du nombre des vehicules en attente donne une meilleure convergence de la fonction; cependant, l'algorithme imite le mutualisme, le commensalisme et la relation parasitaire pour améliorer la qualité d'une fonction objective.

D'après les résultats obtenus, nous remarquons qu'il y'a un perfectionnement dans chaque exécution d'un mécanisme. Nous expliquons cette amélioration des performances du DSOS par les mécanismes de bénéfice mutuel et du vecteur parasite qui sont uniques au DSOS.

Le mécanisme de facteur de bénéfice mutuel dans la phase de mutualisme confère au processus de recherche un pouvoir d'exploitation en lui permettant de parcourir les meilleures régions de solution.

le mécanisme de commensalisme permet d'explorer de nouvelles solutions sans affecter négativement la solution actuelle. Cela signifie que l'algorithme peut continuer à rechercher de meilleures solutions tout en conservant une bonne solution à chaque itération.

Le mécanisme de commensalisme permet également de diversifier l'espace de recherche, ce qui peut aider à éviter de rester bloqué dans des optimas locaux. Dans l'ensemble, le mécanisme de commensalisme permet d'équilibrer l'exploration et l'exploitation, ce qui se traduit par de meilleures performances et une optimisation plus efficace.

La technique du vecteur parasite en phase de parasitisme est capable d'empêcher une convergence prématurée en éliminant les solutions inactives et en introduisant une solution plus active qui éloigne les processus de recherche des optimas locaux.

La phase de parasitisme confère au processus de recherche la capacité d'exploration en ne se concentrant pas uniquement sur les meilleures régions de solution qui pourraient probablement piéger la recherche dans une certaine région de

recherche. Les mécanismes jouent un rôle vital dans l'exploration et l'exploitation dans le processus de recherche.

DSOS a moins de paramètres et plus facile à mettre en œuvre, ce qui est considéré comme un avantage en plus de l'explorabilité et de l'exploitabilité. La méthode est capable d'améliorer sa qualité du processus de recherche, ce qui signifie que DSOS a une probabilité plus élevée d'obtenir une solution quasi optimale.

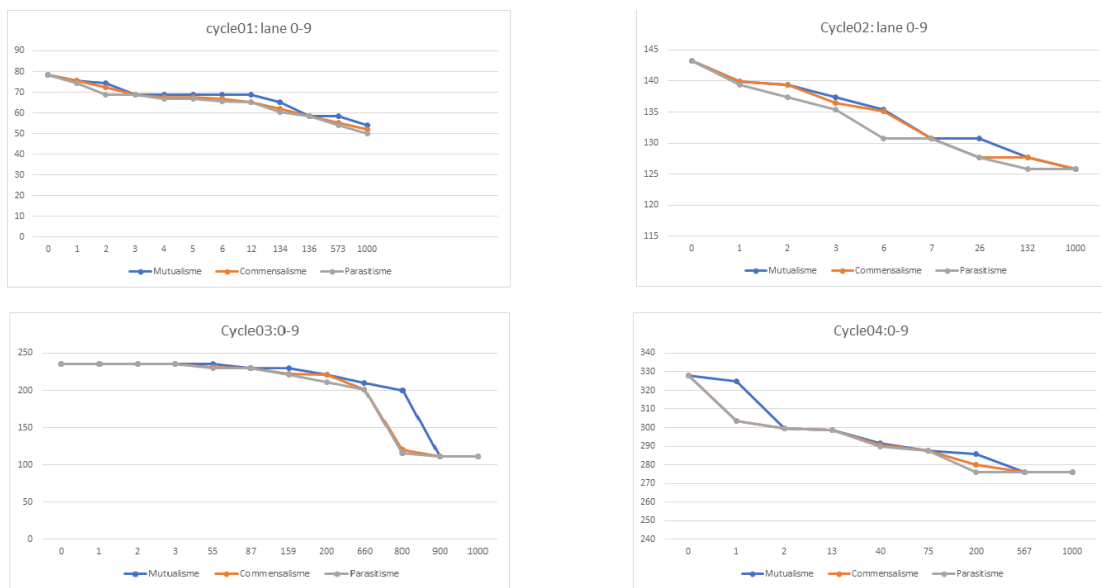


FIGURE 4.9 – Comparaison des Résultats de la synchronisation du DSOS.

## 4.5. CONCLUSION

Dans ce chapitre, nous avons présenté et analysé un certain nombre d'expériences pour mettre en évidence le comportement de nos approches et comparer les résultats obtenus avec l'approche statique (temps fixe). Nous avons réalisé plusieurs simulations en jouant sur différents paramètres. Nous avons étudié l'impact de ces paramètres sur la métrique du nombre de véhicules en attente. L'analyse des résultats obtenus nous a permis de prouver que nos approches permettent d'améliorer les performances du système en réduisant le nombre de voitures en attente.

# CONCLUSION ET PERSPECTIVES

Dans cette thèse, nous avons proposé deux approches. La première approche est une procédure d'optimisation par les AG pour améliorer la fluidité du trafic et réduire la congestion dans les zones urbaines.

En utilisant l'algorithme génétique pour l'optimisation des feux de signalisation, il est possible d'explorer efficacement un large espace de solutions et de rechercher des plans de signalisation optimaux. Les opérateurs de sélection, de croisement et de mutation de l'algorithme génétique permettent d'explorer différentes combinaisons de phases de feux de signalisation, de durées et de synchronisations pour trouver des solutions qui minimisent les retards et réduisent les congestions.

L'avantage clé de l'algorithme génétique est sa capacité à itérer et à améliorer progressivement les solutions au fil du temps. En évaluant la qualité des solutions à l'aide d'une fonction d'adaptation appropriée, l'algorithme génétique peut sélectionner les meilleurs individus et combiner leurs caractéristiques les plus performantes pour générer des solutions de plus en plus optimales.

L'optimisation des feux de signalisation basée sur l'algorithme génétique est une approche prometteuse pour améliorer la gestion du trafic urbain. En exploitant les principes de l'évolution naturelle, cet algorithme permet de rechercher des solutions efficaces en termes de synchronisation des feux de signalisation. Cependant, une analyse approfondie des caractéristiques spécifiques du réseau routier, des conditions de trafic et des objectifs d'optimisation est nécessaire pour adapter et paramétrer l'algorithme génétique de manière optimale.

La deuxième approche est dédiée à la stratégie discret symbiotic organism search. DSOS offre une solution novatrice pour améliorer la fluidité du trafic et réduire la congestion dans les zones urbaines. DSOS s'inspire des relations symbiotiques observées dans la nature pour résoudre les problèmes complexes de synchronisation des feux de signalisation.

Avec l'algorithme DSOS est possible de maintenir une diversité de solutions et de favoriser la coopération entre elles. Les organismes dans DSOS échangent

## Conclusion et Perspectives

---

des informations et s'adaptent en fonction du succès des autres, ce qui permet une exploration plus efficace de l'espace des solutions et une convergence vers des solutions de meilleure qualité.

L'avantage clé de DSOS est sa capacité à exploiter la coopération entre les solutions. Les échanges d'informations entre les organismes symbiotiques permettent de partager les connaissances et les expériences pour améliorer les performances globales du système de signalisation. Cela permet de trouver des plans de signalisation qui minimisent le nombre des véhicules en attente.

DSOS permet d'explorer efficacement l'espace des solutions et de converger vers des plans de signalisation optimaux.

Les résultats des différentes simulations réalisées montrent que les performances obtenues par nos approches sont plus élevées par rapport aux méthodes statiques. Ainsi nous pouvons dire que l'objectif de notre contribution qui consistait à minimiser la métrique définie a été atteint.

## PERSPECTIVES

Certaines pistes de recherche sont possibles pour améliorer notre proposition :

- Optimisation en temps réel adaptative : Il s'agit d'appliquer les algorithmes d'optimisation en temps réel adaptatifs afin d'ajuster en continu les durées des feux en fonction des conditions de trafic changeantes.
- Intégration avec l'infrastructure des villes intelligentes : étudiez l'intégration des algorithmes d'optimisation des feux de signalisation avec d'autres infrastructures des villes intelligentes, telles que les systèmes de transport intelligents, les centres de gestion du trafic et les outils d'urbanisme.
- Optimisation multi-objectif : considérez plusieurs objectifs simultanément, tels que la minimisation du temps de trajet, la réduction des émissions et l'amélioration de la sécurité des piétons. Développez des algorithmes d'optimisation multi-objectif capables de trouver des compromis entre des objectifs contradictoires et de fournir une gamme de solutions Pareto-optimales.

## BIBLIOGRAPHIE

- [1] Gao, K., Zhang, Y., Sadollah, A., Su, R. (2016). Optimizing urban traffic light scheduling problem using harmony search with ensemble of local search. *Applied Soft Computing*, 48 :359-372.
- [2] Abushehab, R. K., Abdalhaq, B. K., Sartawi, B. (2014). Genetic vs. particle swarm optimization techniques for traffic light signals timing. 2014 6th International Conference on Computer Science and Information Technology (CSIT).
- [3] Kumar, A., Sing, RR. (2017). Traffic congestion and possible solution in urban transport system, 4th International Conference on Emerging Trends in Engineering Technology, Science and Management,603-607.
- [4] Bazzan, C., Anna L. (2007). Traffic as a Complex System : Four Challenges for Computer Science and Engineering.
- [5] Hu, W., Wang, H., Yan, L., Du, B. (2015). A swarm intelligent method for traffic light scheduling : application to real urban traffic networks. *Applied Intelligence*, 44 : 208-231.
- [6] Xie, X., Smith, S. F., Chen, T., Barlow, G. J. (2014). Real-time traffic control for sustainable urban living. 17th International IEEE Conference on Intelligent Transportation Systems (ITSC).
- [7] Sklenar, J., Beranek, Z., Popela, P. (2009). Simulation-based heuristic optimization of A traffic systems. ECMS 2009 Proceedings Edited by J. Otamendi, A. Bargiela, J. L. Montes, L. M. Doncel Pedrera, 315321.
- [8] Adacher, L. (2012). A global optimization approach to solve the traffic signal synchronization problem. *Procedia, Social and Behavioral Sciences*, 54 : 12701277.
- [9] Garey, M.r. and Johnson, D.s. (1979) *Computers and Intractability A Guide to the Theory of NP-Completeness*. scientific research publishing.
- [10] Beheshti, Z., Shamsuddin, S. M. (2013). A review of population-based meta-heuristic algorithm. *International Journal of Advances in Soft Computing and Its Applications*. 5 : 1-35.
- [11] Torres-Jiménez, J., Pavón, J. (2014). Applications of metaheuristics in real-life problems. *Progress in Artificial Intelligence*, 2 : 175176
- [12] Faris, H., Mafarja, M. M., Heidari, A. A., Aljarah, I., Al-Zoubi, A. M., Mirjalili, S., Fujita, H. (2018). An efficient binary Salp Swarm Algorithm with crossover scheme for feature selection problems. *Knowledge-Based Systems*, 154 : 4367



## Bibliographie

---

- [13] Garcia-Nieto, J., Olivera, A. C., Alba, E. (2013). Optimal Cycle Program of Traffic Lights With Particle Swarm Optimization. *IEEE Transactions on Evolutionary Computation*, 17 : 823-839.
- [14] Sevaux, M. (2004). Métaheuristiques : Stratégies pour l'optimisation de la production de biens et de services. (Metaheuristics : strategies for the optimisation of the production of goods and services).
- [15] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in 2009 World congress on nature biologically inspired computing (NaBIC), 2009, pp. 210-214.
- [16] Rao, R. V. (2016a). Review of applications of TLBO algorithm and a tutorial for beginners to solve the unconstrained and constrained optimization problems. *Decision Science Letters*, 1-30.
- [17] Rao, R. V., Patel, V. (2012). An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems. *International Journal of Industrial Engineering Computations*, 3 : 535-560.
- [18] Wolpert, D., Macready, W. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1 : 67-82.
- [19] Askerzade, I. N., Mahmood, M. (2010). "Control the extension time of traffic light in single junction by using fuzzy logic". *International Journal of Electrical Computer Sciences IJECSIJENS*, 10 : 48-55.
- [20] Greenshields, B. D., Bibbins, J., Channing, W. and Miller, H. (1935), A study of traffic capacity, in 'Highway research board proceedings', Vol. 14, pp. 448-477.
- [21] Foy, M. D., Benekohal, R. F., Goldberg, D. E. (1992). Signal timing determination using genetic algorithms. *Transportation Research Record*, (1365), 108.
- [22] Ceylan, H., Bell, M. G. (2004). Traffic signal timing optimisation based on genetic algorithm approach, including drivers' routing. *Transportation Research Part B : Methodological*, 38 : 329-342.
- [23] Sáez, D., Cortés, C. E., Núñez, A. (2008). Hybrid adaptive predictive control for the multi-vehicle dynamic pick-up and delivery problem based on genetic algorithms and fuzzy clustering. *Computers Operations Research*, 35 : 3412-3438.
- [24] Uzunova, M. (2009). Commande non-entière des systèmes : développement et application pour les modèles du flux de trafic routier (Doctoral dissertation, Artois).

## Bibliographie

---

- [25] Heydecker, B. and Addison, J. (2011), Analysis and modelling of traffic flow under variable speed limits, *Transportation research part C : Emerging Technologies* 19 : 206-217.
- [26] Singh, G., Bansal, D., Sofat, S. (2014). Intelligent transportation system for developing countries—a survey. *International Journal of Computer Applications*, 85- 34-38.
- [27] Raza, A., Hameed, S., Macintyre, T. (2008). Global positioning system working and its applications. In *Innovations and Advanced Techniques in Systems, Computing Sciences and Software Engineering* (pp. 448-453). Springer Netherlands.
- [28] Zhang, W. (2019). Planning and evaluation of autonomous vehicles in freight and public transport services (Doctoral dissertation, KTH Royal Institute of Technology).
- [29] Gupta, M., Benson, J., Patwa, F., Sandhu, R. (2022). Secure V2V and V2I Communication in Intelligent Transportation Using Cloudlets. *IEEE Transactions on Services Computing*, 15(4),19121925.
- [30] Zhang, G., Avery, R. P., Wang, Y. (2007). Video-based vehicle detection and classification system for real-time traffic data collection using uncalibrated video cameras. *Transportation research record*, 1993 : 138-147.
- [31] Sébastien,A.(2009). Capteurs de position innovants : application aux Systèmes de Transport Intelligents dans le cadre d'un observatoire de trajectoires de véhicules. *Institut National Polytechnique de Toulouse*, 25-30.
- [32] Singh, G., Bansal, D., Sofat, S. (2014). Intelligent transportation system for developing countries—a survey. *International Journal of Computer Applications*, 85 : 34-38.
- [33] Chen, Y.-R., Chen, K.-P., Hsiungy, P.-A. (2016). Dynamic traffic light optimization and Control System using model-predictive control method. 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC).
- [34] Chong, E. K., Zak, S. H. (2013). *An introduction to optimization* (Vol. 75). John Wiley Sons.
- [35] Foulds, L. R. (2012). *Optimization techniques : an introduction*. Springer Science Business Media.
- [36] Pedregal, P. (2004). *Introduction to optimization* (Vol. 46). New York : Springer.
- [37] Rothlauf, F. (2013). *Design of modern heuristics : Principles and application*. Berlin : Springer

## Bibliographie

---

- [38] Arockia, S. (2017). Nature Inspired Metaheuristic Algorithms. *International-Research Journal of Engineering and Technology (IRJET)*, 04(10).
- [39] Rao, R. V., Patel, V. (2012). An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems. *International Journal of Industrial Engineering Computations*, 3 : 535-560.
- [40] Neumüller, C., Wagner, S., Kronberger, G., Affenzeller, M. (2012). Parameter Meta-optimization of Metaheuristic Optimization Algorithms. *Computer Aided Systems Theory EUROCAST 2011 Lecture Notes in Computer Science*, 367-374.
- [41] Rao, R., Savsani, V., Vakharia, D. (2011). Teaching-learning-based optimization : A novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, 43 : 303-315.
- [42] Rao, R. V. (2016). Review of applications of TLBO algorithm and a tutorial for beginners to solve the unconstrained and constrained optimization problems. *Decision Science Letters*, 1-30.
- [43] Kar, A. K., (2016). Bio inspired computing A review of algorithms and scope of applications. *Expert Systems With Applications*, Issue 59, pp. 20-32
- [44] Monmarché, N. (2000). Algorithmes de fourmis artificielles : applications à la classification et à l'optimisation (Doctoral dissertation, Université François Rabelais-Tours).
- [45] Dorigo, M., Birattari, M., Stutzle, T. (2006). Ant colony optimization. *IEEE computational intelligence magazine*, 1 : 28-39.
- [46] Jabbarpour, M. R. et al., 2014. Ant colony optimisation for vehicle traffic systems : applications and challenges. *International Journal of Bio-Inspired Computation*, 6(1), pp. 32-56.
- [47] Rothlauf, F. (2011). *Design of modern heuristics : principles and application* (Vol. 8, No. 9). Heidelberg : Springer.
- [48] Kar, A. K., 2016. Bio inspired computing A review of algorithms and scope of applications. *Expert Systems With Applications*, Issue 59, pp. 20-32.
- [49] Maniezzo, V. Trubian, M., 1994. Ant system for job-shop scheduling. *Belgian Journal of Operations Research*, 34(1), pp. 39-54
- [50] Ugur, A. Aydin, D., 2009. An interactive simulation and analysis software for solving TSP using Ant Colony Optimization algorithms. *Advances in Engineering Software*, 40(5), p. 341349.
- [51] Claes, R. Holvoet, T., 2012. *Cooperative Ant Colony Optimization in Traffic Route Calculations*. Springer Berlin Heidelberg, pp. 23-34.

## Bibliographie

---

- [52] Anon., 1995. Particle swarm optimization. *Neural Networks, 1995. Proceedings., IEEE International Conference on*, Volume 14, pp. 1942-1948.
- [53] Poli, R. (2008). Analysis of the publications on the applications of particle swarm optimisation. *Journal of Artificial Evolution and Applications*, 110.
- [54] Beni, G. Wang, J., (1993). *Swarm Intelligence in Cellular Robotic Systems. Robots and Biological Systems : Towards a New Bionics*, Volume 102, pp. 703-712.
- [55] García-Nieto, J., Alba, E. Carolina Olivera, A., (2012). Swarm intelligence for traffic light scheduling : Application to real urban areas. *Engineering Applications of Artificial Intelligence*, 25(2), pp. 274-283.
- [56] D. Karaboga, B. Akay, and C. Ozturk (2007), Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks. in *International conference on modeling decisions for artificial intelligence*, pp. 318-329.
- [57] D. Karaboga and B. Basturk (2008). On the performance of artificial bee colony (ABC) algorithm, *Applied soft computing*, vol. 8, no. 1, pp. 687-697.
- [58] D. Karaboga and B. Basturk (2007), A powerful and efficient algorithm for numerical function optimization : artificial bee colony (ABC) algorithm. *Journal of global optimization*, vol. 39, no. 3, pp. 459-471.
- [59] Karaboga, D. (2010). Artificial bee colony algorithm. *Scholarpedia Journal*, 5(3), 6915.
- [60] He, J. Hou, Z., 2012. Ant colony algorithm for traffic signal timing optimization. *Advances in Engineering Software*, 43(1), pp. 14-18.
- [61] Yuan, S., Zhao, X. An, Y., 2014. Identification and optimization of traffic bottleneck with signal timing. *Journal of Traffic and Transportation Engineering*, 1(5), Pages 353 -361.
- [62] K. Gao, Yicheng Zhang, A. Sadollah and Rong Su (2017), Improved artificial bee colony algorithm for solving urban traffic light scheduling problem, *IEEE Congress on Evolutionary Computation (CEC)*, Donostia, Spain, pp. 395-402.
- [63] Adebiji, R.F. ; Abubilal, K.A. ; Muazu, M.B. ; Adebiji, B.H.(2018). Development and simulation of adaptive traffic light controller using artificial bee colony algorithm. *Int. J. Intell. Syst. Appl.* , 10, 6874.
- [64] X.S. Yang and S. Deb (2009), "Cuckoo search via Lévy flights," in *2009 World congress on nature biologically inspired computing (NaBIC)*, pp. 210-214.
- [65] X.S. Yang and S. Deb (2014), "Cuckoo search : recent advances and applications," *Neural Computing and Applications*, vol. 24, no.1, pp. 169-174.
- [66] X.S. Yang and S. Deb (2014), *Cuckoo Search. In Nature-Inspired Optimization Algorithms* (pp. 129139). Elsevier.

## Bibliographie

---

- [67] S. Araghi, A. Khosravi, and D. Creighton (2015), "Design of an optimal ANFIS traffic signal controller by using cuckoo search for an isolated intersection," in 2015 IEEE international conference on systems, man, and cybernetics, pp. 2078-2083.
- [68] S. Araghi, A. Khosravi, and D. Creighton (2015), "Intelligent cuckoo search optimized traffic signal controllers for multi-intersection network," *Expert Systems with Applications*, vol. 42, no. 9, pp. 431-442.
- [69] X.-S. Yang (2010), "A new metaheuristic bat-inspired algorithm," in *Nature inspired cooperative strategies for optimization (NICSO 2010)*, Springer, pp. 65-74.
- [70] A. H. Gandomi, X.-S. Yang, A. H. Alavi, and S. Talatahari (2013), "Bat algorithm for constrained optimization tasks," *Neural Computing and Applications*, vol. 22, no. 6, pp. 1239-1255.
- [71] Yahya Zebari, A., M. Almufti, S., Mohammed Abdulrahman, C. (2020). Bat algorithm (BA) : review, applications and modifications. *International Journal of Scientific World*, 8(1),pp 1.
- [72] S. Srivastava and S. K. Sahana (2019), "Application of bat algorithm for transport network design problem," *Applied Computational Intelligence and soft computing*.
- [73] K. Jintamuttha, B. Watanapa, and N. Charoenkitkarn (2016), "Dynamic traffic light timing optimization model using bat algorithm," in 2016 2nd International Conference on Control Science and Systems Engineering (ICCSSE), pp. 181-185.
- [74] Srivastava, S.; Sahana, S.K (2019). Bat Algorithm-Based Traffic Signal Optimization Problem. In *Soft Computing for Problem Solving, Socpros 2017* ; Springer : Singapore, Volume 1, pp. 927936.
- [75] K. Jintamuttha, B. Watanapa, and N. Charoenkitkarn (2016), "Dynamic traffic light timing optimization model using bat algorithm," in 2016 2nd International Conference on Control Science and Systems Engineering (ICCSSE), pp. 181-185.
- [76] L. N. De Castro and J. Timmis (2002), "An artificial immune network for multi-modal function optimization," in *Proceedings of the Congress on Evolutionary Computation. CEC'02*, vol. 1, pp. 699-704.
- [77] Syahputra, R., Soesanti, I. (2017). An artificial immune system algorithm approach for reconfiguring distribution network.
- [78] B. Trabelsi, S. Elkosantini, and S. Darmoul (2012), "Traffic Control at Intersections Using Artificial Immune System Approach," 9th International Conference of Modeling, Optimization and Simulation, MOSIM'12.

## Bibliographie

---

- [79] S. Darmoul, S. Elkosantini, A. Louati, and L. B. Said (2017), "Multi-agent immune networks to control interrupted flow at signalized intersections," *Transportation Research Part C : Emerging Technologies*, vol. 82, pp. 290-313.
- [80] D. Moalla, S. Elkosantini, and S. Darmoul(2013), "An artificial immune network to control traffic at a single intersection," in *Proceedings of 2013 International Conference on Industrial Engineering and Systems Management (IESM)*, pp. 1-7.
- [81] R. Galvan-Correa et al.(2020), "Micro Artificial Immune System for Traffic Light Control," *Applied Sciences*, vol. 10, no. 21, p. 7933.
- [82] S. Mirjalili, S. M. Mirjalili, and A. Lewis (2014), "Grey wolf optimizer," *Advances in engineering software*, vol. 69, pp. 46-61.
- [83] X. Zhang, Q. Lin, W. Mao, S. Liu, Z. Dou, and G. Liu (2014), "Hybrid Particle Swarm and Grey Wolf Optimizer and its application to clustering optimization," *Applied Soft Computing*, vol. 101, p. 61-107.
- [84] T.-C. Teng, M.-C. Chiang, and C.-S. Yang (2019), "A hybrid algorithm based on GWO and GOA for cycle traffic light timing optimization," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pp. 774-779
- [85] S. S. Sabry and N. M. Kaittan (2020), "Grey wolf optimizer based fuzzy-PI active queue management design for network congestion avoidance," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, no. 1, pp. 199-208.
- [86] Z. W. Geem, J. H. Kim, and G. V. Loganathan(2001), "A new heuristic optimization algorithm : harmony search," *simulation*, vol. 76, no. 2, pp. 60-68.
- [87] H. Ceylan and H. Ceylan (2012), "A hybrid harmony search and TRANSYT hill climbing algorithm for signalized stochastic equilibrium transportation networks," *Transportation Research Part C : Emerging Technologies*, vol. 25, pp. 152-167.
- [88] K. Gao, Y. Zhang, A. Sadollah, and R. Su (2016), "Optimizing urban traffic light scheduling problem using harmony search with ensemble of local search," *Applied Soft Computing*, vol. 48, pp. 359-372.
- [89] M. Dell'Orco, O. Baskan, and M. Marinelli(2013), "A Harmony Search Algorithm approach for optimizing traffic signal timings," *PROMET-TrafficTransportation*, vol. 25, no. 4, pp. 349-358.
- [90] D. P. Bertsekas, and D. P. Bertsekas (1995), *Dynamic programming and optimal control*, Athena Scientific, vol. 1, no. 2.

## Bibliographie

---

- [91] D. Stefanoiu, P. Borne, D. Popescu, F. G. Filip, and A. El-kamel (2014), Optimization in engineering sciences : Metaheuristic, stochastic methods and decision support, Wiley.
- [92] E. L. Lawler and D. E. Wood (1966), Branch-and-bound methods : A survey, Operations research, vol. 14, no. 4, pp. 699719.
- [93] F. Yan, M. Dridi, and A. E. Moudni (2008), Control of traffic lights in intersection : A new branch and bound approach, in 2008 International Conference on Service Systems and Service Management, Melbourne, Australia, pp. 16.
- [94] G. B. Dantzig (1998), Linear programming and extensions, Princeton university press.
- [95] D. Bertsimas and J. N. Tsitsiklis (1997), Introduction to linear optimization, Athena Scientific, vol. 6.
- [96] Garey and Johnson (1979), Computers and Intractability : a guide to the theory of NP-completeness.
- [97] Digabel, S. L. (2014). Introduction aux métaheuristiques. Technical report, Ecole Polytechnique de Montreal.
- [98] Dréo, J., Pétrowski, A., Siarry, P., Taillard, E. (2003). Métaheuristiques pour l'optimisation difficile (p. 356). Eyrolles.
- [99] Jaoua, M. (2014). Algorithme de recherche tabou pour la planification optimale d'une campagne marketing sur les moteurs de recherche (Doctoral dissertation, École Polytechnique de Montréal).
- [100] Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, MA.
- [101] Dorigo, M. and Stutzle, T. (2004). Ant Colony Optimization. MIT Press, Cambridge,MA.
- [102] Hassel, O. (2023). Shortest path routing in a road network : Finding an easily implementable algorithm given efficiency constraints. Diva-portal.org.
- [103] Goffinet, François(2018). Introduction au protocole de routage dynamique OSPF. cisco.goffinet.org.
- [104] Holland, D. E., Olesen, R. J., Bevins, J. E. (2021). Multi-objective genetic algorithm optimization of a directionally sensitive radiation detection system using a surrogate transport model. Engineering Applications of Artificial Intelligence, 104, 104357.
- [105] Burjorjee KM (2010). Generative fixation : A unified explanation for the adaptive capacity of simple recombinative genetic algorithms. SIGEVolution, 4 :1213.

- [106] Whitley D, Starkweather T, and Bogart C (1990). Genetic algorithms and neural networks : optimizing connections and connectivity. *Parallel Computing*, 14(3) :347-361.
- [107] Goldberg, D. E (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, Reading, Massachusetts.
- [108] Whitley, D(1994). A genetic algorithm tutorial, *Statistics and Computing*, vol. 4, , p.p. 65-85.
- [109] Cheng, M.-Y., Prayogo, D. (2014). Symbiotic Organisms Search : A new metaheuristic optimization algorithm. *Computers Structures*, 139, 98112.
- [110] Rossi, F. (1997). *Initiation à la programmation Java*. Université Paris-IX Dauphine.
- [111] Rochdi Bachir Bouiadjra (2015). *Commande  $H_\infty$  non linéaire*, thèse de doctorat en sciences, Université Ahmed Bel bella, Oran1.
- [112] Poulain, P. F.et all. Introduction - Cours de Python. Univ-paris-diderot.fr , from [https://python.sdv.univ-paris-diderot.fr/01\\_introduction/](https://python.sdv.univ-paris-diderot.fr/01_introduction/)
- [113] Behrisch M, Bieker L, Erdmann J, Krajzewicz D. SUMO - Simulation of Urban Mobility, An Overview. In *The Third International Conference on Advances in System Simulation*; 2011 ; Berlin : Institute of Transportation Systems, German Aerospace Center. p. 1.
- [114] Krajzewicz D, Hertkorn G, Wagner P. SUMO (Simulation of Urban MObility) An open-source traffic simulation. Berlin : German Aerospace Centre.



## TABLE DES FIGURES

1.1	<i>Headway.</i>	21
1.2	<i>Débit de véhicules (Q).</i>	22
1.3	<i>Densité.</i>	22
1.4	<i>Schéma fondamental du trafic.</i>	23
1.5	<i>Indice d'ordre des véhicules dans les modèles microscopiques..</i>	25
1.6	<i>Système de transport intelligent.</i>	27
1.7	<i>Véhicule autonome.</i>	29
1.8	<i>Communication V2V et Communication V2I</i>	30
1.9	<i>Capteur vidéo</i>	31
2.1	<i>Classification des algorithmes métaheuristiques.</i>	40
2.2	<i>Illustration d'optimisation d'essaim de particules.</i>	43
2.3	<i>Oiseau de coucou.</i>	48
2.4	<i>Principe général du système immunitaire artificiel.</i>	52
2.5	<i>Les loups gris (GWO).</i>	54
3.1	<i>Architecture générale de l'approche DJ-GA.</i>	66
3.2	<i>Exemple de la première stratégie.</i>	67
3.3	<i>Représentation du chromosome.</i>	75
3.4	<i>Processus de Croisement.</i>	76
3.5	<i>Processus de Mutation.</i>	77
3.6	<i>L'organigramme d'Adaptation d'Algorithme Génétique .</i>	77
3.7	<i>Architecture générale de l'approche(ACO-DSOS).</i>	79
3.8	<i>Application de l'algorithme Ant Colony.</i>	81
3.9	<i>Organigramme d'Adaptation d'algorithme symbiotic organism search.</i>	88
4.1	<i>Simulation of Urban MObility (SUMO).</i>	92
4.2	<i>Interface graphique du SUMO.</i>	93
4.3	<i>Sélection de la zone étudiée.</i>	95

## Liste des Figures

---

4.4	<i>Visualisation du scénario créé par osmWebWizard . . . . .</i>	96
4.5	<i>Structure globale pour le traitement de la zone étudiée. . . . .</i>	97
4.6	Comparaison des Résultat de la synchronisation GA avec Temps fixe.	102
4.7	Comparaison des Résultat de la synchronisation du GA avec Temps fixe. . . . .	103
4.8	Comparaison des Résultat de la synchronisation du DSOS avec Temps fixe. . . . .	106
4.9	Comparaison des Résultat de la synchronisation du DSOS. . . . .	108

## LISTE DES TABLEAUX

2.1	Travaux Connexes sur les Algorithmes de gestion de trafic basé sur ACO. . . . .	42
2.2	Travaux Connexes sur les Algorithmes de gestion de trafic basé sur PSO. . . . .	45
2.3	Travaux Connexes sur les Algorithmes de gestion de trafic basé sur ABC. . . . .	47
2.4	Travaux connexes sur les algorithmes de gestion de trafic basé sur CS. . . . .	49
2.5	Travaux connexes sur les algorithmes de gestion de trafic basé sur BA. . . . .	51
2.6	Travaux connexes sur les algorithmes de gestion de trafic basé sur AIS. . . . .	53
2.7	Travaux connexes sur les algorithmes de gestion de trafic basé sur GWO. . . . .	55
2.8	Travaux connexes sur les algorithmes de gestion de trafic basé sur HS. . . . .	56
3.1	Paramètres de la fonction objectif. . . . .	69
4.1	Paquet SUMO. . . . .	94
4.2	Résultat de la synchronisation du signal sur les AG. . . . .	100
4.3	Résultat de la synchronisation du signal de trafic sur temps fixe. . . . .	101