

*République Algérienne Démocratique et Populaire*  
*Ministère de l'enseignement supérieur et de la recherche scientifique*

*Université Mohamed Seddik Benyahia Jijel*  
*Faculté des sciences exactes et d'informatique*  
*Département d'informatique*



**Mémoire de Fin d'Etude pour l'obtention du Diplôme de  
Master en Informatique**

**Option**

**Réseaux et sécurité**

**Thème**

**Développement d'un système  
intelligent de gestion des feux de  
circulation à Base des algorithmes  
génétiques**

**Encadré Par :**

**Dr. Souici Ismahane**

**Réalisé par :**

**Aibeche Younes**

**Promotion : 2020.**

# ★Remerciements★

*je remercie, en premier lieu, notre Dieu le très  
miséricordieux qui a bien voulu nous donner la force et le  
courage pour effectuer le présent travail.*

*je exprime nos remerciements à notre encadreur M<sup>me</sup> .Souici Ismahane pour  
l'assistance qu'elle je ai témoignée, pour sa disponibilité, pour ces orientations et  
conseils sans lesquels ce travail ne verra pas le jour, qu'elle trouve ici l'expression  
de notre gratitude.*

*À ma familles pour leurs rôle principale, leurs encouragement  
et leurs soutien de me avoir guidés à arriver à ce que je suis  
maintenant.*

*Je remerciai également, tous les enseignants qui ont assurés mon formation  
durant mon cycle universitaire de Jijel.*

*Je remerciai les membres de jury qui ont accepté de jurer mon travail. Qu'ils  
trouvent ici, l'expression de notre profond, respect et notre sincère gratitude.*

*À toutes les personnes qui ont participés de près ou de loin à  
la réalisation de ce travail.*

*J'ai l'honneur de dédier ce travail à tous ceux,*

*À tous, un grand merci*

# *★Dédicaces★*

*Je dédie ce travail*

*A ma mère Nora, la femme la plus chère du monde, la source de tendresse qui a tout donné sans rien recevoir, je le remercie du fond de mon cœur.*

*A mon père Houcin , qui a toujours cru en moi, et a mis à ma disposition tous les moyens nécessaires pour que je réussisse dans mes études.*

*A mes chères sœurs .*

*A mes chers frères .*

*A toute ma famille.*

*A tous mes chères amies*

*A mes collègues Master 2 Réseaux et Sécurité.*

*A tous qui ont contribué de près ou de loin à la réalisation de ce travail.*

## ملخص

بدأ مفهوم المدينة الذكية في أوائل التسعينيات مع ظهور تقنيات جديدة للشبكات المحمولة واللاسلكية ، بالإضافة إلى التطور الهائل لتقنيات الإنترنت مثل الويب الدلالي وإنترنت الأشياء . يعتبر نظام النقل الذكي أحد التطبيقات الرئيسية للمدينة الذكية. فمع الزيادة الهائلة والسريعة في عدد المركبات تنزايد مشكلة الازدحام المروري في المناطق الحضرية حول العالم. نتيجة لذلك ، فإن إضاعة الوقت على الطريق مجرد الانتظار ، وإهدار وقود أكثر تكلفة على أنشطة غير منتجة تظل العواقب الأكثر إزعاجًا. في هذا العمل نقترح نظام ذكي لإدارة إشارات المرور من خلال استغلال الخوارزميات الجينية ، وهو نظام ديناميكي يتكيف مع تقلبات الطريق من أجل تقليل الازدحام المروري في المدن.

**كلمات مفتاحية:** المدينة الذكية ، النقل الذكي ، الازدحام ، إشارات المرور الذكية ، الخوارزميات الجينية.

# Résumé

Le concept de ville intelligente a commencé au début des années 1990 avec l'émergence de nouvelles technologies de mobiles et de réseaux sans fil, en plus de l'énorme développement des technologies Internet telles que le web sémantique et l'Internet des objets (IoT, Internet of Things). Le système de transport intelligent est considéré comme l'une des applications majeures de la ville intelligente. Avec l'accroissement immense et rapide du nombre de véhicules, le problème de congestion routière s'accroît dans les zones urbaines à travers le monde. De ce fait, une perte de temps sur la route pour ne faire qu'attendre, et un gaspillage de carburant plus coûteux pour des activités non productives, restent les conséquences les plus gênantes. Dans ce travail, nous proposons un système intelligent de gestion de feux de circulation en exploitant des algorithmes génétiques, et qui soit dynamique et qui s'adapte aux fluctuations de la route dans le but de réduire les congestions routières dans les villes.

**Mots clés :** Ville intelligente, Transport intelligent, Congestion, Feux de circulation intelligent, Algorithmes génétiques.

## Abstract

The smart city concept began in the early 1990s with the emergence of new mobile and wireless network technologies, in addition to the enormous development of Internet technologies such as the semantic web and the Internet of Things (IoT , Internet of Things). The intelligent transport system is considered one of the major applications of the smart city. With the huge and rapid increase in the number of vehicles, the problem of road congestion is growing in urban areas around the world. As a result, wasting time on the road just waiting, and wasting more expensive fuel on unproductive activities, remain the most annoying consequences. In this work, we propose an intelligent traffic light management system by exploiting genetic algorithms, and which is dynamic and which adapts to the fluctuations of the road in order to reduce traffic congestion in cities.

**Keywords:** Smart city, Smart transportation, Congestion, Smart traffic lights, Genetic algorithms.

# *Sommaire*

<b>Résumé.....</b>	<b>I</b>
<b>Liste des figures.....</b>	<b>II</b>
<b>Liste des tableaux.....</b>	<b>III</b>
<b>Liste des algorithmes.....</b>	<b>IV</b>
<b>Introduction générale.....</b>	<b>1</b>

## **CHAPITRE 1 : SYSTEMES DE TRANSPORT INTELLIGENTS**

1. Introduction.....	4
2. Congestion du trafic.....	4
2.1. Définition de la congestion .....	4
2.2 .Types de congestion.....	5
2.2.1.Congestion récurrent.....	5
2.2.2.Congestion non-récurrent .....	5
2.3.Impacts négatifs.....	5
3. Systèmes de transports intelligents (STI).....	6
3.1. Application des STI à l’exploitation des réseaux routiers .....	6
3.2. Contexte et applications au milieu urbain .....	7
4. Systèmes de gestion de trafic routier.....	8
5. Outils théoriques.....	10
5.1. Contrôle par logique floue.....	10
5.2.Réseaux de neurones.....	11
5.3 Théorie des files d’attente....	12
5.4. Algorithmes génétiques .....	13
6. Conclusion .....	14

---

## CHAPITRE 2 : ALGORITHMES GENETIQUES

1. Introduction.....	15
2. Historique.....	15
3. Présentation.....	17
4. Paramètres d'un AG.....	19
5. Principe de base d'un AG standard.....	20
5.1. Le codage.....	22
5.2. La sélection.....	24
5.2.1. La loterie biaisée.....	24
5.2.2. La méthode élitiste.....	25
5.2.2. La sélection par tournoi.....	25
5.2.4. La sélection universelle stochastique (SUS).....	25
5.3. Le croisement.....	26
5.3.1. Croisement multipoints.....	27
5.3.2. Croisement uniforme.....	28
5.4. Mutation.....	29
5.4.1. Insertion.....	29
5.4.2. Echange (1-Opte ou Swap mutation).....	30
5.4.3. Inversion.....	30
5.4.4. $\lambda$ -Opt.....	30
5.4.5. Or-Opt.....	30
5.4.6 Mélange (Scramble Mutation ou SM).....	31
5.5. Critère d'arrêt.....	31
6. Avantages des AGs.....	32
7. Conclusion.....	32

---



## CHAPITRE 3 : CONCEPTION

1. Introduction.....	33
2. Formalisation du problème de chiffrement.....	34
3. Feu de circulation génétique.....	36
3.1. Le codage.....	36
3.2 La création de la population initiale .....	36
3.3. Opérateurs de reproduction .....	37
3.3.1. Croisement .....	37
3.3.2. Mutation .....	39
3.4. L'évaluation des individus .....	39
3.5.La sélection des individus .....	40
3.6. Le critère d'arrêt.....	42
4. Points forts des algorithmes génétiques.....	42
5.Conclusion .....	42

---

## CHAPITRE 4 : EXPERIMENTATION ET RESULTATS

1. Introduction.....	43
2. L'environnement de programmation JAVA.....	44
3. La structure du programme.....	44
4.Paramètres génétiques et illustrations.....	45
4.1. Paramètres génétique.....	45
4.2. Illustrations .....	46
4.2.1 Codage des chromosomes.....	46
4.2.2.Génération de la population initiale.....	46
4.2.3.Opération de reproduction.....	47
4.2.3.1.Croisement.....	47
4.2.3.1.Mutation.....	48
5. Présentation des interfaces.....	49
5.1. Fenêtre d'accueil.....	49
5.2. Fenêtre de simulation.....	49
6. Résultats expérimentaux et discussion.....	50
6.1. Résultats expérimentaux.....	50
6.1..1Exemple 1.....	50
6.1.2.Exemple 2.....	52
6.1. discussion.....	53
8. Conclusion.....	53
<b>Conclusion générale.....</b>	<b>54</b>
<b>Bibliographie.....</b>	<b>55</b>

## *Liste des figures*

<b>Figure1.1:</b> Réseau " ordonné " sans connexion arrière .....	12
<b>Figure2.1:</b> L'organigramme d'un AG standard.....	21
<b>Figure2.2:</b> Les cinq niveaux d'organisation d'un AG.....	23
<b>Figure2. 3:</b> Illustration schématique du codage des variables réelles.....	24
<b>Figure2. 4:</b> La méthode de sélection de la loterie biaisée.....	25
<b>Figure2.5:</b> Sélection universelle stochastique.....	26
<b>Figure2.6:</b> Croisement avec un point.....	27
<b>Figure2.7:</b> Croisement avec 2 points.....	28
<b>Figure2. 8:</b> Illustration du croisement uniforme.....	29
<b>Figure2.9:</b> Opérateur d'insertion, échange et inversion.....	30
<b>Figure2.10:</b> Or-opt.....	31
<b>Figure2.11:</b> Exemple de scramble Mutation.....	31
<b>Figure3.1</b> Intersection entre deux routes principales.....	34
<b>Figure3.2</b> La proposition de quatre intersections.....	35
<b>Figure3.3</b> Codage adapté.....	36
<b>Figure3.4</b> Exemple sur du croisement à deux points.....	38
<b>Figure 4.1</b> Exemple d'initialisation du temps des feux verts .....	46
<b>Figure 4.2</b> Exemple de chromosomes générés à partir du chromosome initial illustré par la figure 4.1.....	46
<b>Figure 4.3</b> Exemple d'enfants calculés par croisement entre les parents de la figure 4.2.....	48
<b>Figure 4.4</b> Exemple d'un enfant généré par mutation appliquée sur le chromosome de la figure 4.2.....	48

<b>Figure 4.5</b> La fenêtre d'accueil.....	49
<b>Figure 4.6</b> La fenêtre de simulation.....	50
<b>Figure 4.7</b> Cas de congestion , exemple 1.....	51
<b>Figure 4.8</b> L'état e la route suite à la réponse du système.....	51
<b>Figure 4.9</b> Cas de congestion , exemple 2.....	52
<b>Figure 4.10</b> L'état e la route suite à la réponse du système.....,	52

## *Liste des tableaux*

<b>Tableau 2.1</b> : Résumé de la terminologie en AG.....	18
<b>Tableau 4.1</b> : Paramètre génétique.....	45

## *Liste des algorithmes*

<b>Algorithme 2.1:</b> Croisement à un point.....	27
<b>Algorithme 2.2:</b> Croisement à deux points.....	28
<b>Algorithme 3.1:</b> Création de la population initiale.....	37
<b>Algorithme 3.2:</b> Algorithme de croisement.....	38
<b>Algorithme 3.3:</b> Algorithme de Mutation.....	39
<b>Algorithme 3.4:</b> Algorithme de la sélection par roulette.....	41

# INTRODUCTION GENERALE

**L**’augmentation constante de la population apporte de nouveaux problèmes aux villes , il fallait construire plus de maisons, de parcs, d’écoles, de bâtiments, de routes, etc. Les embouteillages sont plus fréquents et affectent les citoyens car la qualité de vie diminue en raison du stress, du gaspillage de temps, de l’augmentation de la pollution, des pertes économiques, etc. dans les années à venir, les villes pourraient évoluer dans une croissance chaotique du trafic ou s’appuyer sur la technologie pour améliorer son développement de manière durable. Construire une ville intelligente nécessite la contribution de tous les membres de la communauté, de la municipalité, des chauffeurs et le développement des infrastructures, des services publics, etc., une planification appropriée est également importante pour surmonter les défis auxquels les villes sont confrontées. [1]

L’évolution rapide de la technologie crée des opportunités pour le développement de l’humanité. Les systèmes de transport intelligents sont un concept qui est apparu comme un moyen d’atténuer les problèmes liés à la congestion du trafic. L’infrastructure, les capteurs de véhicules, les caméras, les algorithmes d’IA, l’IoT et les applications peuvent y contribuer.

En effet, l’augmentation de la congestion sur les routes urbaines, en particulier aux heures de pointe, est un problème critique dans de nombreux pays et est devenue une préoccupation majeure pour les spécialistes du transport et des décideurs. Les méthodes existantes pour la gestion du trafic, de surveillance et de contrôle ne sont pas suffisamment efficaces en termes de performance, du coût et de l’effort nécessaire pour l’entretien et le soutien. Cette augmentation de l’urbanisation et de la congestion créent un besoin de faire fonctionner les systèmes de transport avec une efficacité maximale. L’une des mesures pour faire face à ce problème est le contrôle des feux de circulation placés dans des intersections. Un contrôleur de feux de circulation a pour but de réduire les arrêts et les retards de l’ensemble des véhicules.

Jusqu’à présent, la plupart des villes Africaines en général et Algériennes en particulier utilisent des commandes de signalisation classique (à temps fixe) réglées une seule fois tenant compte de prévisions du trafic. On peut observer par exemple que la durée des feux est la même à 8H00 du matin, où la circulation est complètement embouteillée par les camions de marchandises traversant la ville et les particuliers se rendant à leur travail, qu’à 22H00 le soir où la plupart des rues sont complètement dégagées. Ce comportement en soi peut lui même

être générateur de congestion dans la mesure où les commandes d'infrastructure ne s'adaptent pas aux fluctuations du trafic. On voit bien là, la nécessité de systèmes plus dynamiques, qui ont fait l'objet ces dernières années d'une attention particulière. On parle alors des Systèmes de Transport Intelligents (En anglais : Intelligent transportation Systems (ITS)).

De manière schématique, un système de contrôle dynamique de trafic dispose des commandes permettant de contrôler le trafic et des capteurs lui donnant des informations sur l'état du trafic : les nombres de véhicules sur certaines rues ou intersections du réseau routier. Les informations obtenues sont traitées par un ordinateur qui a pour mission de donner aux commandes les valeurs (durée et synchronisation des feux) permettant de minimiser une certaine fonction (la longueur des files d'attente aux intersections, du temps d'attente moyen aux feux, du temps d'attente moyen de la plus longue file d'attente, etc.) exprimant la congestion du réseau routier. Dans un système de contrôle dynamique, la durée des feux serait donc recalculée à intervalle régulier dans le souci de minimiser la congestion [2].

De plus, l'intelligence artificielle est un domaine de l'informatique qui a eu une pénétration exceptionnelle dans les activités de la vie réelle comme le trading, les jeux, la prédiction, la finance, la robotique, l'optimisation, etc. L'IA semble être fiable lorsqu'il s'agit de résoudre les problèmes de congestion du trafic; de nombreuses approches d'IA telles que la logique floue, les agents intelligents, l'apprentissage automatique, les réseaux de neurones artificiels, la vision artificielle, les algorithmes génétiques, etc., peuvent être implantées pour lutter contre la congestion du trafic.

Les algorithmes dits génétiques (ou évolutionnaires) représentent une importante catégorie de ces méthodes. Leurs domaines d'application ne cessent pas de s'élargir vu leurs multiples points forts (performance, rapidité, simplicité de leurs opérateurs...), où ils connaissent un grand succès surtout dans le domaine de l'intelligence artificielle et recherche opérationnelle pour résoudre des problèmes d'optimisation. En plus leur large exploitation du caractère aléatoire dans la majorité des étapes de leur processus, encourage leur exploitation dans le domaine des feux de circulations en donnant lieu à un système dynamique adaptable.

C'est d'ailleurs l'objet de notre présent travail à travers lequel nous cherchons à développer un système intelligent de gestion des feux de circulation.

Ainsi et hormis cette introduction et la conclusion générale qui reprennent les travaux de l'ensemble des chapitres, le manuscrit est divisé en quatre chapitres.



En effet Dans le premier chapitre, nous avons présenté la congestion routière et la gestion du trafic routier en général et les système intelligents pour la régulation ainsi que les outils théoriques qui sont utilisés pour la modélisation du trafic routier. Nous avons, en particulier, mis l'accent sur la théorie des algorithmes génétiques que nous avons utilisée pour modéliser notre système.

De son tour, le deuxième chapitre présente une introduction aux algorithmes génétiques en décrivant le vocabulaire de ce domaine, les différentes étapes du processus génétique ainsi que les techniques de base utilisées pour la conception d'un algorithme génétique simple, robuste et efficace. Tout en citant les avantages et les inconvénients de ces algorithmes.

Dans le troisième chapitre, nous proposons l'application des algorithmes génétiques pour résoudre le problème de gestion des feux de circulations, où nous décrivons les différentes étapes de conception de notre modèle proposant une façons intelligente pour gérer les feux de circulation.

Au niveau du quatrième chapitre, nous détaillerons le choix paramétrique de l'algorithme proposé et nous rapporterons des exemples de résultats obtenus par notre système développé.

# CHAPITRE 1

## Systemes de transport intelligents

### 1.Introduction

**D**e nos jours, se déplacer est devenu un aspect essentiel de la vie quotidienne .De cette raison le transport routier devient un élément fondamental dans le fonctionnement des sociétés modernes. En effet, des millions de véhicules transportent des personnes et des marchandises sur les réseaux routiers chaque jour. La gestion d'un tel réseau est devenue un élément du base . Dans ce chapitre nous présentons des généralités sur la congestion urbaine, les systèmes de transport intelligents, et leurs applications pour gérer la circulation routière.

### 2.Congestion du trafic

La congestion urbaine apparaît lorsque la demande de déplacements excède la capacité de l'infrastructure .Cette dernière est citée comme un mal majeur et grandissant dans les milieux urbains. De nombreux conducteurs se retrouvent chaque jour dans la congestion et les embouteillages, bien que son importance et sa fréquence varient d'un endroit à l'autre.

Les autorités urbaines, régionales et provinciales de transport s'appliquent à gérer cette problématique au moyen d'une vaste gamme de mesures. Individuellement, les conducteurs n'interprètent pas tous de la même façon le sens de la congestion et n'ont pas la même perception. Pour certains, il s'agit d'un véritable embouteillage avec des arrêts complets et de longs retards. Pour d'autres, il s'agit plutôt d'une situation où l'on avance lentement, ou plus généralement, où on se déplace à une vitesse inférieure à la limite permise.

#### 2.1 Définition de la congestion

La congestion est un phénomène physique concernant la façon dont des véhicules empêchent la progression des uns et des autres au fur et à mesure que la demande d'un espace routier s'approche de la capacité maximale de celui-ci de même qu'un phénomène relatif ayant trait aux attentes des usagers vis-à-vis des performances d'un réseau routier. Donc La congestion est une situation dans laquelle la demande d'espace routier dépasse l'offre.[3].

## 2.2 Types de congestion

Selon les causes, la congestion peut être classée en deux catégories : congestion récurrente ou congestion non-récurrente [4].

### 2.2.1 Congestion récurrente

La congestion récurrente est due à l'insuffisance de l'offre par rapport à la demande dans des conditions normales. Comme le cycle de vie d'une infrastructure est assez long, la capacité nominale d'une section routière, qui est calculée en fonction d'une projection lors de la conception, pourrait être inférieure à la demande réelle à un moment ultérieur. Même si la capacité conçue satisfait un niveau moyen de la demande, ce qui est le cas typique, la congestion récurrente paraît pendant les périodes de pointe : les périodes de pointe peuvent être les heures de pointe pour un réseau routier/autoroutier urbain/périurbain, les périodes de vacances pour les grands axes de liaison ou d'autres occasions plus spécifiques.

### 2.2.2 Congestion non- récurrente

La congestion non-récurrente est due aux perturbations sur le réseau qui réduisent éventuellement les capacités comme les incidents, les accidents, les travaux sur la route ou réduisent les vitesses pratiquées par les usagers comme de mauvaises conditions météorologiques, et probablement d'autres événements spéciaux. Certaines perturbations sont prévisibles, par exemple les travaux programmés, d'autres sont moins prévisibles comme de mauvaises conditions météorologiques, voire complètement imprévisibles comme les incidents, les accidents. La part de la congestion non-récurrente dans l'ensemble de la congestion varie en fonction de l'endroit mais elle est toujours dans un ordre de grandeur considérable, en particulier pour les grandes agglomérations.

## 2.3 Impacts négatifs

La congestion du trafic a plusieurs effets négatifs [5]:

- Les embouteillages obligent un automobiliste à perdre son temps sur la route dans un manière productive.
- En raison de la congestion du trafic, une énorme quantité de produits pétroliers est gaspillée rien.

- La congestion du trafic augmente également la pollution atmosphérique et sonore dans une zone affecte finalement la santé des automobilistes.
- La congestion de la circulation augmente également l'effet de serre dans une zone, de sorte que l'électricité la consommation augmente dans les maisons le long d'une route pour les garder au frais.
- Usure des véhicules due au ralenti dans la circulation et aux accélérations fréquentes et le freinage, ce qui entraîne des réparations et des remplacements plus fréquents.
- La congestion de la circulation peut également entraver la trajectoire des véhicules d'urgence.
- Cela augmente également le stress et la frustration d'un automobiliste et donc les cas de rage au volant augmente.

### **3.Systèmes de transports intelligents (STI)**

Les systèmes de transports intelligents (STI) fait référence aux efforts visant à ajouter des technologies de l'information et des communications aux infrastructures de transport ou aux véhicules. Les premiers systèmes ITS, mis en œuvre bien avant l'introduction du terme ITS, se présentaient sous la forme de contrôleurs de trafic (également appelés feux de signalisation ou robots). L'objectif principal de ces contrôleurs était d'améliorer les flux de trafic et, en général, on peut conclure que les premiers systèmes STI ont été explorés lorsque les ingénieurs du trafic se sont rendu compte qu'ils ne pouvaient pas se sortir de la congestion. On espère que l'utilisation des technologies de la communication améliorera l'utilisation de la capacité routière existante [6].

#### **3.1 Application des STI à l'exploitation des réseaux routiers**

Les principales fonctions d'exploitation du réseau liées aux STI sont les suivantes :

- Surveillance du réseau, maintien de la viabilité et de la sécurité routières, contrôle du trafic, information d'aide aux déplacements et aux usagers et gestion de la demande. La surveillance du réseau, est une des fonctions clés de la collecte d'information courante sur le réseau routier et du soutien des autres activités d'exploitation du réseau. Par conséquent, même si la surveillance fait partie intégrante de tous les services de STI, elle ne constitue habituellement pas un service en soi. Afin d'améliorer la viabilité et la sécurité routières, des mesures proactives

et réactives peuvent être déployées. Les mesures proactives mettent la priorité sur la prévention des incidents et de la congestion, et les mesures réactives mettent la priorité sur la détection et la vérification des incidents et des conditions routières dangereuses, les interventions et le dégagement, et sur le rétablissement des conditions normales.

- L'information aux voyageurs est une mesure de prévention secondaire. La diffusion d'avertissements en temps opportun, quant aux conditions de circulation dangereuses et à la congestion, réduit le nombre d'accidents.

- La sécurité routière ne peut être assurée uniquement qu'au moyen de mesures préventives. Il est important de surveiller constamment l'état de la route même et de la circulation, et de réagir aux situations.

### **3.2 Contexte et applications au milieu urbain**

Les STI sont appliqués aussi dans le milieu urbain dans :

- **Les intersections**

Le champ d'application des STI en milieu urbain est très large : en premier lieu, ces derniers agissent sur les intersections, en se chargeant d'appliquer une stratégie de changement des feux de circulation. Cette gestion des feux va représenter un aspect essentiel de la fluidité du trafic routier dans une ville [7].

- **Le stationnement**

La gestion du stationnement est également primordiale et a une incidence directe sur la fluidité du trafic : il paraît logique de dire que l'utilisation de la voiture en milieu urbain repose en partie sur le fait de savoir s'il y a une place disponible ou non sur le lieu d'arrivée. Il faut savoir qu'en moyenne, en France par exemple, 10% des véhicules en circulation à un instant donné cherchent une place où se garer. Les STI vont aider à prendre des décisions, mais également informer les utilisateurs ou encore contrôler les véhicules [7].

Par Exemple, des détecteurs peuvent être utilisés afin de détecter la présence d'un véhicule sur une place, et calculer sa durée de stationnement. Ceci a été constaté dans la ville d'Amiens via des stationnements "minute" : une borne est associée à une place, et dès lors qu'un véhicule s'y gare, un compte à rebours se déclenche pour une durée déterminée. Si cette durée est dépassée, les agents de la voie publique sont automatiquement prévenus. Nous pouvons également citer l'utilisation de panneaux à messages variables (PMV) pour les parkings, systèmes très répandus

dans les grandes métropoles qui indiquent le nombre de places disponibles (ceci n'utilisant pas nécessairement des détecteurs, mais étant généralement calculé en fonction des entrées / sorties dans le parking en lui-même).

- **La sécurité routière**

Les STI Visent à réduire le risque d'accidents de voiture et à minimiser les dommages résultant d'accidents inévitables. Ces applications imposent des exigences, nécessitant un matériel fiable dédié ainsi que des communications fiables et opportunes. Ces applications comprennent la sensibilisation des coopératives, par exemple les applications de gestion des progrès, l'avertisseur de sortie de voie et à la gestion de la vitesse, ainsi que les applications de détresse, par exemple la détection des dangers et les mauvaises conditions météorologiques[8].

- **Les ronds-points**

Certains travaux abordent la gestion des ronds-points, en se basant sur les priorités existantes. Dans de nombreux pays, il est, par exemple, commun de donner la priorité aux usagers étant déjà à l'intérieur d'un rond-point.

Ce qui en ressort apparaît comme une évidence : soit deux flots, chacun étant sur une voie respectivement au cœur du rond-point et sur les côtés, la priorité doit être donnée au flot de la plus grande importance afin d'obtenir une gestion optimale du trafic.

Les feux quant à eux améliorent grandement la fluidité du trafic, mais uniquement en cas de débit suffisamment élevé [7].

#### **4. Systèmes de gestion de trafic routier**

Les avancées ont été permises grâce à l'introduction de plusieurs solutions novatrices en termes de gestion du trafic routier : au total, ce sont plus d'une vingtaine de projets qui sont nés durant ces trente dernières années [2].

Dans cette partie, nous allons essentiellement parler de deux systèmes ne sont pas nécessairement les plus performants, mais représentent les deux principaux systèmes de gestion du trafic routier utilisés dans le monde [9]:

- **SCOOT** [10]

SCOOT (Split Cycle Offset Optimisation Technique) un système de contrôle à la fois réactif et adaptatif et entièrement centralisé, développé par le TRL (Traffic research laboratory, un centre de recherche anglais sur les transports). Dans ce système, l'ensemble des informations recueillies sur le terrain vont à un centre de gestion, qui s'occupe de traiter les informations et renvoyer des indications directement aux intersections. Les véhicules sont détectés par des dispositifs pouvant être placés à divers endroits sur les voies : au niveau des feux ou à une certaine distance afin de pouvoir mesurer le débit du trafic.

SCOOT mesure en permanence le volume de véhicules de chaque côté de l'intersection et change la durée des phases en fonction d'un indice de performance, calculé par rapport au délai d'attente

moyen, de la longueur des files d'attente et des arrêts sur le réseau. Ainsi, de la même manière que TRANSYT [11], système sur lequel il est basé, SCOOT génère des plans de feux en fonction de la demande des utilisateurs (côté adaptatif). En plus de cela, le système utilise des informations en ligne issues de bases de données (historique ou autres, côté réactif).

- **SCATS** [12]

SCATS (Sydney Coordinated Adaptive Traffic System) qui a été à l'origine développé pour Sydney et d'autres villes Australiennes. Il est pour sa part entièrement adaptatif et utilise une notion d'hierarchie (ce qui forme une certaine distribution sur le réseau). Entre le recueil des données sur le terrain et le centre de contrôle, des contrôleurs intermédiaires sont insérés, permettant d'alléger la charge globale du système et d'avoir un contrôle découpé en plusieurs zones, l'ensemble des acteurs utilisant des communications synchronisées. De manière similaire à SCOOT, ce système ajuste le temps des cycles et autres paramètres en fonction des données recueillies afin de diminuer le délai et les arrêts, mais n'utilise pas la même stratégie. Les valeurs recueillies permettent la sélection de plans de feux parmi une large librairie, sur lesquels le système va se baser pour proposer des plans adaptés. De plus, contrairement à SCOOT, les détecteurs sont uniquement placés au niveau des feux de circulation.

Selon [13], l'installation de l'une de ces solutions prendrait en moyenne 365 heures (630h pour SCOOT), coûterait au total en moyenne 55.000\$ par intersection, et nécessiterait un temps d'entraînement moyen de 41 heures (dont 60h pour SCATS).

Tandis que SCATS et SCOOT sont conçus afin de ne prendre qu'une décision par cycle, d'autres systèmes plus modernes tels que OPAC [14], RHODES [15] ou encore InSync[10]

analysent le trafic en temps réel et prennent des décisions seconde après seconde afin d'adapter dynamiquement un cycle. Nous pouvons également citer CRONOS [16] et Prodyn [17], les deux principaux systèmes agissant sur les carrefours Français [18]. Le premier a été mis au point au début des années 1990 par l'INRETS et permet de dynamiser un carrefour en se basant notamment sur des images en temps réel en provenance de caméras, qui fournissent des informations telles que l'occupation de la chaussée. Le deuxième a été développé par le CERT et a la particularité de se baser sur des mécanismes classiques de cycles et phases.

De nombreux autres systèmes de ce type existent, nous n'avons cité que les principaux. Ces solutions "classiques" régissent à elles seules une bonne partie de ce qu'il peut exister en termes de gestion du trafic routier de manière complète et dynamique. Hélas, leur coût et leur durée d'installation constituent un frein à leur développement.

## 5. Outils théoriques

Il est courant pour des modèles dynamiques de la littérature se servir d'outils théoriques, parfois en faisant un rapprochement à la réalité, parfois sans aucune notion physique (technologie utilisée, disposition). Dans les systèmes de gestion de trafic, plusieurs outils théorique sont étudiés, répandus dans la littérature et servant de base à certains modèles étudiés.

### 5.1 Contrôle par logique floue

La logique floue permet de mettre en place des degrés dans la vérification d'une condition, et non plus se borner à un choix strictement binaire. Ce principe est utilisé par quelques auteurs pour traiter le problème de la gestion des feux de circulation et permet de simplifier le problème, ce qui change des méthodes d'optimisation mathématique habituelles souvent lourdes.

Un exemple de travail peut être trouvé dans [19]. Nous pouvons également citer [20] qui utilise la logique floue afin de déterminer le temps d'un feu en fonction du nombre de véhicules présents sur les voies : à un nombre de véhicules correspond un intervalle définissant une durée de feu (exemple : moins de 5 véhicules par minute donne le feu vert pendant 10 secondes). Ce principe apparaît comme idéal à utiliser :

- Théorie simple s'appliquant à des problèmes complexes.
- Aucun modèle mathématique requis.
- Robustesse de la commande floue par rapport aux incertitudes.



Les inconvénients sont tout de mêmes importants [21] : les techniques de mise en place et les réglages sont empiriques et aucune théorie ne permet de démontrer la stabilité et la robustesse d'une telle méthode.

## 5.2 Réseaux de neurones

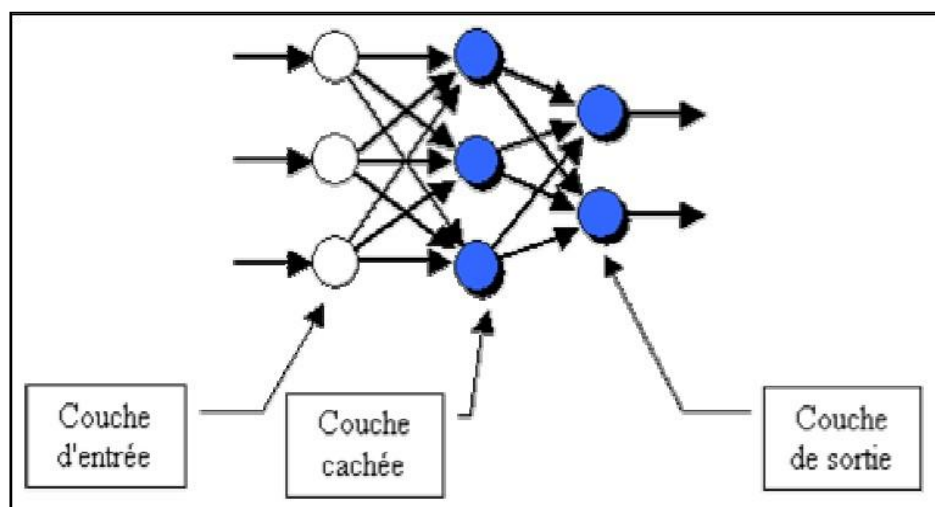
Les réseaux de neurones sont composés d'éléments simples (ou neurones) fonctionnant en parallèle. Ces éléments ont été fortement inspirés du système nerveux biologique. Comme dans la nature, le fonctionnement du réseau (de neurone) est fortement influencé par la connections des éléments entre eux. On peut entraîner un réseau de neurone pour une tâche spécifique (reconnaissance de caractères par exemple).

en ajustant les valeurs des connections (ou poids) entre les éléments (neurone). En général, l'apprentissage des réseaux de neurones est effectué de sorte que pour une entrée particulière présentée au réseau corresponde une cible spécifique. L'ajustement des poids se fait par comparaison entre la réponse du réseau (ou sortie) et la cible, jusqu'à ce que la sortie corresponde. L'apprentissage " par paquet " du réseau consiste à ajuster les poids et biais en présentant les vecteurs d'entrée/sortie de tout le jeu de données.

L'apprentissage " pas à pas ou séquentiel " consiste à ajuster les poids et biais en présentant les composantes du vecteur d'entrée/sortie les unes après les autres. Ce type d'apprentissage est souvent qualifié d'apprentissage " en ligne " ou " adaptatif ".

L'apprentissage permet aux réseaux de neurones de réaliser des tâches complexes dans différents types d'application (classification, identification, reconnaissance de caractères, de la voix, vision, système de contrôle).

Ces réseaux de neurones peuvent souvent apporter une solution simple à des problèmes encore trop complexes ne pouvant être résolus rapidement par les ordinateurs actuels (puissance de calcul insuffisante) ou par notre manque de connaissances. La méthode d'apprentissage dite superviser est souvent utilisée mais des techniques d'apprentissage non supervisé existent pour des réseaux de neurones spécifiques. Ces réseaux peuvent, par exemple, identifier des groupes de données (réseaux de Hopfield).



**FIGURE 1.1** – Réseau " ordonné " sans connexion arrière (réseau feed-forward)

Dans le cas du trafic routier, plusieurs auteurs se sont penchés sur ce schéma . Ici, il est question d'effectuer rapidement des classifications, et d'apprendre à les améliorer, plutôt que de passer par un schéma traditionnel de modélisation. La logique floue et les algorithmes génétiques peuvent être vus comme des compléments aux réseaux de neurones [13].

### 5.3 Théorie des files d'attente

Les origines de la théorie de files d'attente remontent à 1909, lorsque A.K.Erlang en a posé les bases dans ses recherches sur le trafic téléphonique. Elle fait actuellement l'objet de nombreuses publications scientifiques. Plusieurs auteurs ont utilisé la théorie de files d'attente dans différents domaines, par exemple dans [14] ont utilisé cette théorie pour analyser les performances des réseaux en général.

La théorie des files d'attente est particulièrement adaptée au cas de la gestion du trafic routier [22]. Dans le cas des intersections, une file d'attente est automatiquement créée lorsque les véhicules (clients) souhaitent obtenir un feu au vert (serveur). Particulièrement, il est facile avec cette théorie de calculer des valeurs telles que le nombre moyen de véhicules en attente, en service, le temps moyen d'attente ou encore de séjour dans le système. Une file d'attente est un système stochastique composé d'un certain nombre (fini ou non) de place d'attente d'un ou plusieurs serveurs et bien sûr de clients qui arrivent, attendent, se font servir selon des règles de priorité données et quittent le système .

### 5.4 Algorithmes génétiques

Beaucoup de problèmes sont difficilement solubles exactement. La difficulté ne vient pas de la complexité du problème mais de la taille excessive de l'espace des solutions. Il est alors évident de pouvoir les tester toutes pour trouver la meilleure. C'est pourquoi, des heuristiques approchées ont été essayées pour la résolution de tout un tas de problèmes où la recherche systématique n'est pas possible. Ces heuristiques doivent être peu coûteuses en temps machine et doivent permettre de trouver une solution pas trop mauvaise qui justifie leur utilisation à la place d'une recherche par Monte-Carlo (hasard quadrillé) ou par toute autre méthode. Ainsi sont nés beaucoup d'algorithmes comme les algorithmes génétiques.

L'algorithme génétique ne fait que transposer ce que fait la nature à des systèmes artificiels. Il simule les processus évolutifs Darwiniens et génétiques s'appliquant aux chromosomes. Il transforme un ensemble d'objets mathématiques, une population d'individus souvent représentés par des chaînes de caractères pour imiter les chaînes d'ADN, chacun ayant une valeur d'adaptation, en une nouvelle population. L'algorithme fait donc appel à quatre opérateurs de base :

- L'évaluation du niveau d'adaptation d'un individu.
- La sélection : c'est le choix des individus en fonction du niveau d'adaptation.
- Le croisement : c'est le mélange des bagages génétiques.
- La mutation : le bagage génétique est modifié abruptement.

Les auteurs de [23] ou encore [24] ont proposé d'optimiser le temps à une ou plusieurs intersections en se basant sur un algorithme génétique. Le principe est le suivant : obtenir une solution approchée d'un problème d'optimisation lorsqu'il n'existe pas de méthode exacte afin de le résoudre en un temps raisonnable. Dans ce type d'algorithme, la solution est approchée par bonds successifs (mutations). Ainsi, en ayant connaissance du nombre de véhicules et du temps moyen d'attente à une intersection, l'algorithme va pouvoir appliquer des méthodes d'optimisation et les améliorer au fil de sa vie. Si l'idée paraît intéressante, les contraintes sont évidentes pour le cas du trafic routier :

- De nombreux calculs sont nécessaires.
- Paramètres difficiles à déterminer, et il peut y avoir un certain délai avant l'obtention d'un résultat réellement efficace.

## **6.Conclusion**

Dans ce chapitre, nous avons présenté les Systèmes de transports intelligents qui sont conçus pour la régulation de la congestion routière et la gestion du trafic routier en général, ainsi que les outils théoriques qui sont utilisés pour la modélisation du trafic routier en général, et en particulier de gestion de feux de circulation. Le prochain chapitre présente la méthode d'optimisation par algorithmes génétiques que nous allons exploiter par la suite pour proposer notre système de gestion de feu de circulation.

## CHAPITRE 2

# ALGORITHMES GENETIQUES

### 1. Introduction

**L**es algorithmes génétiques (AG) sont des méthodes d'optimisation stochastique s'appuyant sur les techniques dérivées de la génétique et des mécanismes d'évolution de la nature: croisement, mutation, sélection[25].

Les AG empruntent l'essentiel de leur terminologie au modèle évolutionniste : une population évolue de génération en génération pour devenir plus adaptée aux contraintes environnementales. Ses individus, caractérisés par leurs chromosomes, s'accouplent et se reproduisent, selon le principe générale du « struggle for life » (survivance du plus adapté), et l'on observe parfois des mutations [26]. Bien que très réducteurs des points de vue de la biologie, les AG fournissent des mécanismes de recherche Adaptative robustes et puissants[27].

### 2. Historique

C'est en 1860 que Charles Darwin publie son livre intitulé L'origine des espèces au moyen de la sélection naturelle ou la lutte pour l'existence dans la nature. Dans ce livre, Darwin rejette l'existence «de systèmes naturels figés», déjà adaptés pour toujours à toutes les conditions extérieures, et expose sa théorie de l'évolution des espèces: « sous l'influence des contraintes extérieurs, les êtres vivants se sont graduellement adaptés à leur milieu naturel au travers de processus de reproductions ».

Presque simultanément, en 1866, Mendel (« le moine des pois ») publie l'article retraçant dix années d'expériences d'hybridation chez les végétaux (recombinaison des gènes) et l'adresse aux sociétés scientifiques des quatre coins du monde. Les réactions sont mitigées, voire inexistantes. Le monde scientifique n'est pas prêt à reconnaître la qualité de ses résultats.

Ce n'est seulement en 1900, que la publication de trois nouveaux articles signés Hugo de Vries, Carl Correns et Erich von Tschermak.

C'est alors à partir du 20<sup>ème</sup> siècle que la mutation génétique a été mise en évidence par des chercheurs en informatique qui essayent de développer des méthodes permettant aux systèmes d'évoluer de manière normale et efficace face à de nouvelles conditions d'environnement inconnues, variables ou évolutives. Ainsi, les problèmes de traitement de l'information ne seront plus résolus de manière figées, car il ne sera plus indispensable lors de la phase de conception d'un système, d'énumérer toutes les caractéristiques nécessaires pour les conditions d'exploitations connues au moment de la conception [28].

Dans les années 1960, John Holland, ses collègues et ses étudiantes ont mené des recherches, à l'université de Michigan, poussés par deux objectifs principaux [29]:

Premièrement, mettre en évidence et expliquer rigoureusement les processus d'adaptation des systèmes naturels, et deuxièmement, concevoir des systèmes artificiels qui possèdent les propriétés naturels. Toutefois, c'était Bagley qui a mentionné, en premier lieu, l'expression « algorithme génétique » c'était en 1967. Et en 1975 Holland a formulé des algorithmes génétiques : « the canonical genetic algorithm CGA ». Ce modèle a servi de base aux recherches ultérieures et a été, plus particulièrement, repris par David Goldberg qui a publié, en 1989, un ouvrage de vulgarisation des algorithmes génétiques, tout en ajoutant à la théorie des algorithmes génétiques les idées suivantes: un individu est lié à un environnement par son code d'ADN, une solution est liée à un problème par son indice de qualité [26].

L'originalité des travaux de Holland repose en particulier sur le fait qu'il n'a pas considéré seulement les mutations comme mécanisme d'évolution, mais il exploite surtout les mécanismes de croisement, car c'est en croisant les solutions potentielles existant au sein du pool génétique que l'on peut se rapprocher de l'optimum [27].

Trois types d'algorithmes évolutionnaires ont été développés isolément et à peu près simultanément, par différents scientifiques : la programmation évolutionniste de L.Fogel (Californie, USA, 1966), les stratégies d'évolution inventées en 1973 par deux étudiants ingénieurs à l'université de Berlin, I.Rechenberger et H.P.Schwefel et les algorithmes Génétiques de J.Holland en 1975. En 1985 et grâce à Cramer, une autre classe d'algorithmes évolutionnaires a vu le jour. C'est la programmation génétique (PG), considérée en premiers temps comme sous groupe des AG. Mais cette façon de programmer, qui consiste à faire évoluer le code d'un logiciel afin

qu'il remplisse au mieux certaines tâche, a pris son indépendance, en 1992, grâce à J.Kaza (Californie, USA).

Ces différentes classes d'algorithmes évolutionnaires dont les origines différent et qui possèdent chacune leur spécialité, ne diffèrent que sur les détails d'implémentation des opérateurs et sur les procédures de sélection et remplacement de la population. de plus, et malgré que leurs buts soient différents à l'origine, ils ont commencé à sortir de leur isolement et sont maintenant surtout utilisés pour résoudre des problèmes d'optimisation en convergeant vers le modèle unique des algorithmes évolutionnistes qui intègre les particularités de chacun des modèles [28].

### 3. Présentation

Les techniques de recherche et d'optimisation sont, généralement, classées en trois catégories [28]. Énumératives, déterministes et stochastiques. Les AG font partie de la troisième catégorie et quatre caractéristiques les distinguent des autres techniques d'optimisation [29]:

- Ils utilisent un codage des paramètres et non les paramètres eux-mêmes.
- Ils travaillent sur une population d'individus(ou de solutions).
- Ils n'utilisent que les valeurs de la fonction à optimiser, pas sa dérivée, ou une autre connaissance auxiliaire.
- Ils utilisent des règles de transition probabilistes et non déterministes.

En effet, les AG sont des techniques de recherche et d'optimisation stochastique dérivées de la génétique et des mécanismes de la sélection naturelle et de l'évolution. Leurs champs d'application sont très vastes : économie, finance, optimisation de fonctions, planification, et bien d'autres domaines [31].

Les algorithmes génétiques étant basée sur des phénomènes biologiques, il convient de rappeler au préalable quelques termes de génétique et leur définition dans le domaine informatique [32] (tableau 2.1).

Nature	Définition
Individu	Solution à un problème
Population	Ensemble fini d'individus (de solutions).
Adéquation au milieu	Qualité de la solution
Gène	partie élémentaire (caractère) non divisible d'un chromosome.
Chromosome	Solution potentielle du problème sous une forme codée (forme de chaîne de caractères).
Locus	Position dans la chaîne
Allèle	Valeur de caractéristique
Génotype	Structure
Phénotype	Ensemble de paramètres ou une structure décodé
Croissance	Décodage d'une solution depuis sa représentation
Croisement (Cross-over)	Opérateur d'exploration de l'espace de recherche
Mutation	Modification d'une solution
Sélection naturelle	Réutilisation d'une bonne (sous) solution
Fonction de fitness	Qui désigne la fonction d'évaluation d'un individu. Cette fonction est liée à la fonction à optimiser et permet de définir le degré de performance d'un individu (donc d'une solution)

**Tableau2. 1:**Résumé de la terminologie en AG [31], [32].

Les AG sont des méthodes utilisées dans les problèmes d'optimisation qui tirent leur nom de l'évolution biologique des êtres vivants dans le monde réel. Ces algorithmes cherchent à simuler le processus de la sélection naturelle dans un environnement défavorable en s'inspirant de la théorie de l'évolution proposée par C. Darwin. Dans un environnement, « les individus » les mieux adaptés tendent à vivre assez longtemps pour se reproduire alors que les plus faibles ont tendance à disparaître (the survival of the fittest) [33] [34].



Dans un problème d'optimisation à ' $n$ ' variables, nous faisons correspondre un gène à chaque variable cherchée. Chaque gène est représenté par une chaîne de caractères choisis dans un alphabet fini (souvent binaire). Les gènes s'enchaînent ensemble "bout à bout" pour construire un chromosome représentant une solution potentielle sous une forme codée. Ces chromosomes constituent les briques de base contenant les caractéristiques héréditaires des individus. Un chromosome (ou plusieurs) forme un individu qui représente à son tour une solution potentielle dans l'espace de recherche correspondant au problème. Etant donné que les AGs travaillent sur un ensemble de points de l'espace de recherche, nous appelons l'ensemble des points choisis (à savoir les individus) une population. Au fur et à mesure des générations (itérations), une population des individus les mieux adaptés va être créée.

#### 4. Paramètres d'un AG

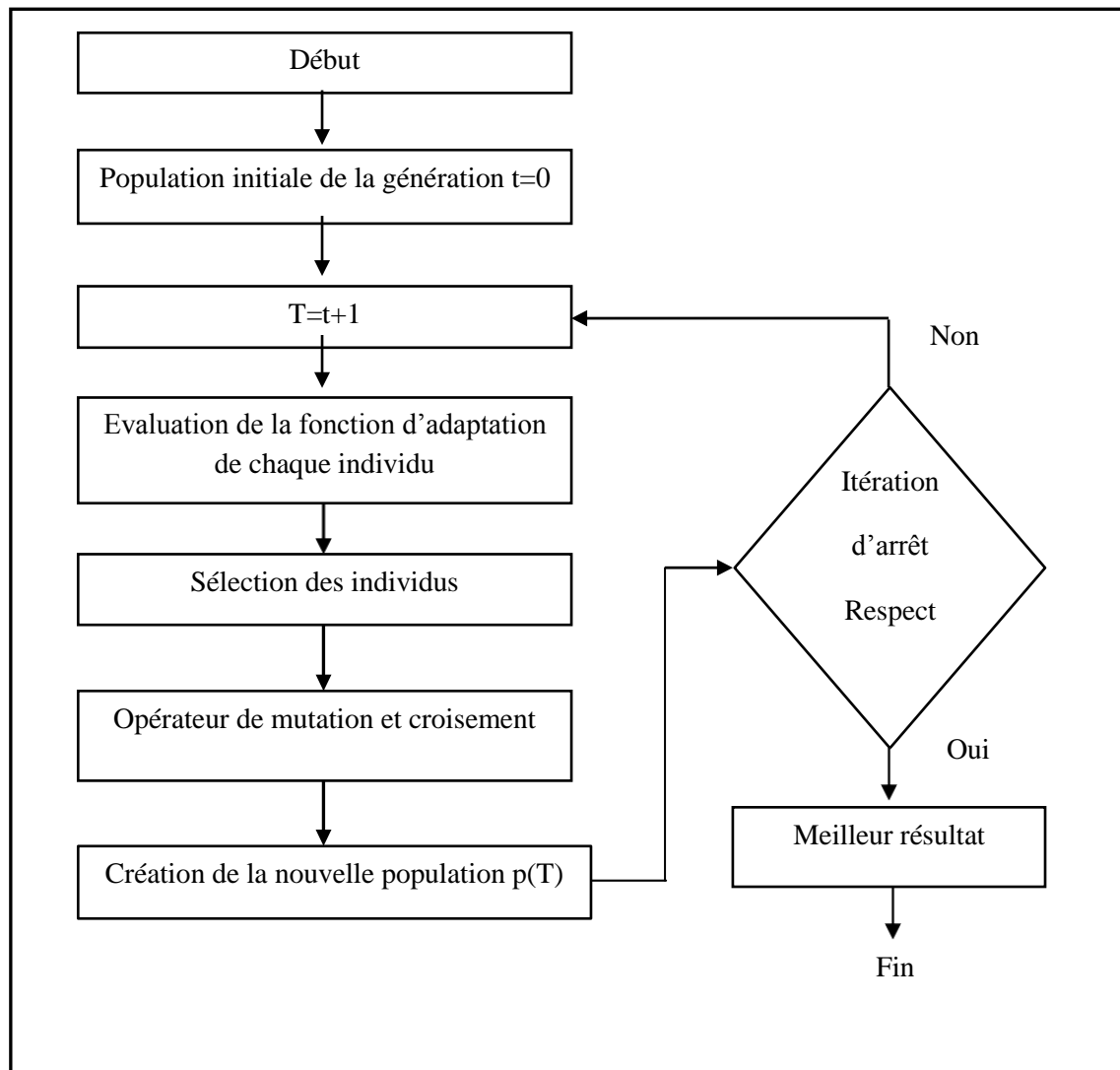
Pour appliquer un AG à un problème réel, on doit posséder les éléments suivants :

- **Un codage des éléments** appartenant à la population, le codage des solutions du problème à résoudre doit être choisi avec soin ;
- **Une fonction d'évaluation ou d'adéquation ou d'adaptation** de l'individu qui mesure la qualité des individus ;
- **Un processus d'évaluation des générations ;**
- **Des opérateurs** pour modifier les individus d'une population de la génération ( $t$ ) à la génération ( $t+1$ ), comme le croisement et la mutation ;
- **Des paramètres de l'AG** les opérateurs précédents dépendent de plusieurs paramètres qui sont fixés à l'avance et dont dépend fortement la convergence de l'algorithme :
- ❖ **Taille de la population** : c'est le nombre d'individus dans la population. Si la taille est trop petite, l'AG peut ne pas converger, par contre si elle est trop grande, l'évaluation des individus peut être très longue ;
- ❖ **Probabilité de croisement et de mutation** : Les valeurs de ces probabilités peuvent varier d'une application à l'autre. Par exemple, dans l'étude des AG pour l'optimisation de cinq fonctions mathématiques, De Jong (1975) a suggéré de choisir une probabilité de croisement élevée, une probabilité de mutation faible (inversement proportionnelle à la taille de la population), et une population de taille modérée. La probabilité de mutation est en général très faible, inférieure à 0,1, une probabilité trop grande, peut modifier les meilleurs individus. [34].
- ❖ **Critère d'arrêt** : c'est le nombre maximal de génération à effectuer.

## 5. Principe de base d'un AG standard

Un AG standard nécessite en premier le codage de l'ensemble des paramètres du problème d'optimisation en une chaîne de longueur finie. Le principe d'un AG est simple, il s'agit de simuler l'évolution d'une population d'individus jusqu'à un critère d'arrêt.

On commence par générer une population initiale d'individus (solutions). Puis, à chaque génération, des individus sont sélectionnés, cette sélection est effectuée à partir d'une fonction objectif appelée fonction d'adaptation. Puis, les opérateurs de croisement et de mutation sont appliqués et une nouvelle population est créée. Ce processus est itéré jusqu'à un critère d'arrêt. Le critère le plus couramment utilisé est le nombre maximal de générations que l'on désire effectuer. La Figure 2.1 présente le principe de l'AG standard. L'AG débute par la génération d'une population initiale et l'évaluation de la fonction d'adaptation de tous les individus qui composent cette première population. Puis, des individus sont sélectionnés aléatoirement pour la reproduction selon le principe de la survie du plus adapté. Ensuite, des individus « enfants » (ou des descendants) sont générés en appliquant les deux opérateurs génétiques suivants : le croisement et la mutation. Ces enfants sont placés dans une nouvelle population  $P(t)$  et vont se substituer, en tout ou en partie, à la population de la génération précédente. De nouvelles populations d'individus vont ensuite se succéder, d'une génération  $(t)$  à la génération  $(t+1)$ , jusqu'à l'atteinte du critère d'arrêt.



**Figure2.1:**L'organigramme d'un AG standard [33].

L'ensemble des étapes suivantes [26] présente une lecture du contenu de l'organigramme de la figure 2.1.

Etape0 : Définir un codage du problème

Etape1 : Créer une population initiale  $p_0$  de  $q$  individus  $\{x_1, x_2, \dots, x_q\}$

$I := 0$  ;

Etape2 : Evaluation des individus.

Soit  $F$  la fonction d'évaluation. Calculer  $F(x_i)$  pour chaque individu  $x_i$  de  $p_i$

Etape3 : Sélection

Sélectionner les meilleurs individus (au sens de  $F$ ) et les grouper par paire.

Etape4 : Application des opérateurs génétiques

1- croisement : appliquer l'opération de croisement aux paires sélectionnées

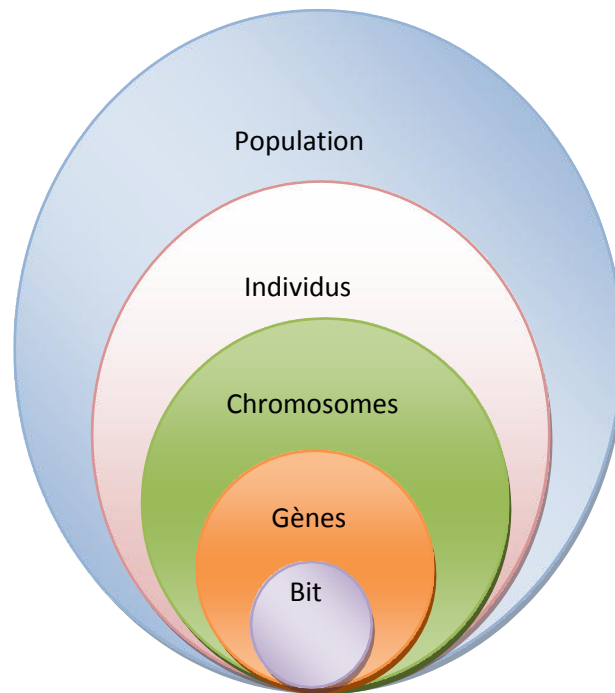
2- mutation : appliquer la mutation aux individus issus du croisement

Ranger les nouveaux individus obtenus (de 1 et 2) dans une nouvelle génération  $p_{i+1}$ ,

Répéter les étapes 2, 3 et 4 jusqu'à l'obtention du niveau de performance souhaité

### 5.1 Le codage

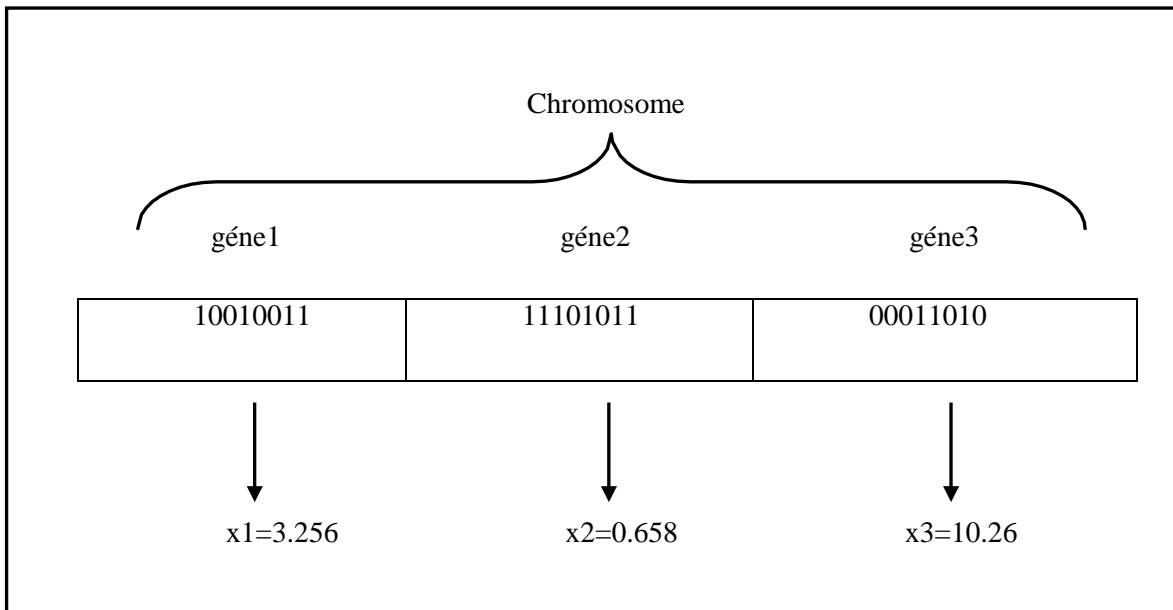
Chaque paramètre d'une solution est assimilé à un gène, toutes les valeurs qu'il peut prendre sont les allèles de ce gène, on doit trouver une manière de coder chaque allèle différent de façon unique (établir une bijection entre l'allèle "réel" et sa représentation codée). Un chromosome est une suite de gène, on peut par exemple choisir de regrouper les paramètres similaires dans un même chromosome (chromosome à un seul brin) et chaque gène sera repérable par sa position : son locus sur le chromosome en question. Chaque individu est représenté par un ensemble de chromosomes, et une population est un ensemble d'individus. [25].



**Figure2.2:**Les cinq niveaux d'organisation d'un AG.

Il y a deux principaux types de codage utilisables, et on peut passer de l'un à l'autre, plus ou moins, facilement :

- ❖ **Le codage binaire** : c'est le plus utilisé des types de codage. Chaque gène dispose du même alphabet binaire  $\{0, 1\}$ . Un gène est alors représenté par un entier long (32 bits), les chromosomes qui sont des suites de gènes sont représentés par des tableaux de gènes et les individus de notre espace de recherche sont représentés par des tableaux de chromosomes. Ce cas peut être généralisé à tout alphabet allélique *n-aire* permettant un codage plus intuitif. Par exemple, pour le problème du voyageur de commerce on peut préférer utiliser l'alphabet allélique  $\{c_1, c_2, c_3, \dots, c_n\}$  où  $c_i$  représente la ville de numéro  $i$  ;
- ❖ **Le codage réel** : cela peut-être utile notamment dans le cas où l'on recherche le maximum ou le minimum d'une fonction réelle.



**Figure2. 3:** Illustration schématique du codage des variables réelles.

## 5.2 La sélection

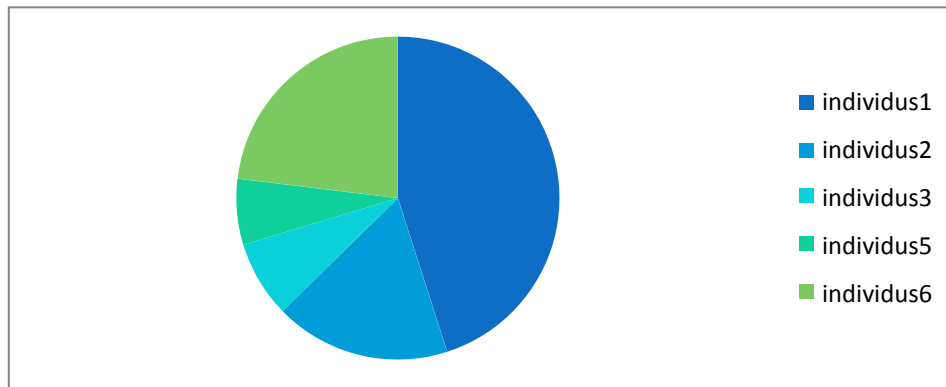
Cet opérateur est chargé de définir quels seront les individus de la population  $P$  qui vont être dupliqués dans la nouvelle population  $P'$  et vont servir de parents (application de l'opérateur de croisement). Soit  $n$  le nombre d'individus de  $P$ , on doit en sélectionner  $n/2$  (l'opérateur de croisement nous permet de repasser à  $n$  individus). Cet opérateur est peut-être le plus important puisqu'il permet aux individus d'une population de survivre, de se reproduire ou de mourir. En règle générale, la probabilité de survie d'un individu sera directement reliée à son efficacité relative au sein de la population. On trouve essentiellement des méthodes de Sélection différentes :

- La méthode de la "loterie biaisée" (roulette Wheel) de Goldberg.
- La méthode "élitiste".
- La sélection par tournois.
- La sélection universelle stochastique.

### 5.2.1 La loterie biaisée

Elle est encore dite roulette Wheel. Elle s'applique comme suit :

A chaque individu est associé un secteur de la roulette dont la largeur est proportionnelle à l'adéquation de cet individu. On sélectionne les « survivants » en tirant au hasard  $N$  individus à l'aide de cette roulette ( $N$  : taille de la population).



**Figure2. 4:**La méthode de sélection de la loterie biaisée.

### 5.2.2 La méthode élitiste

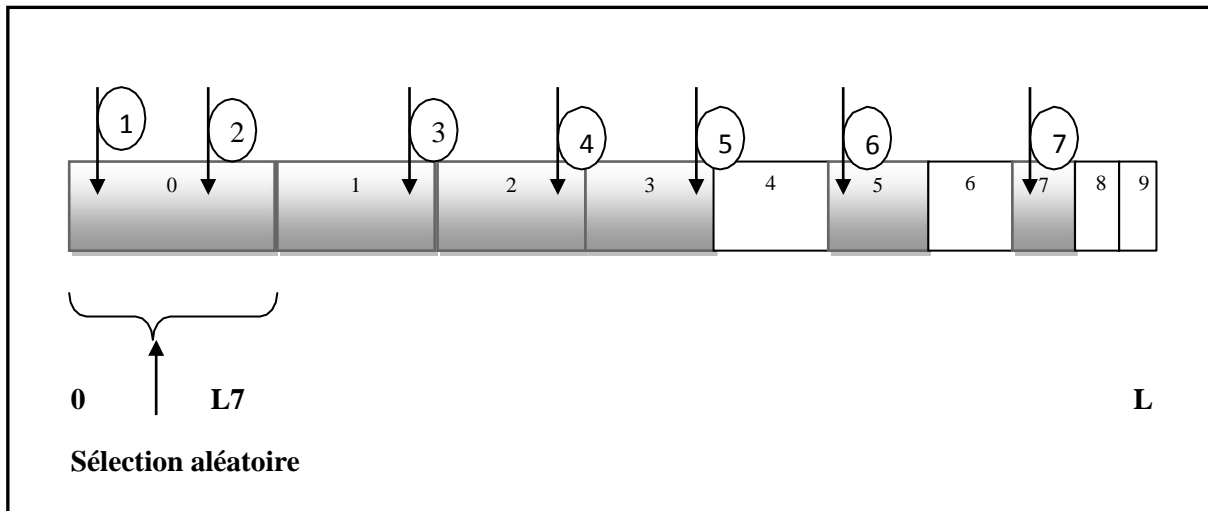
Cette méthode consiste à sélectionner les  $n$  individus dont on a besoin pour la nouvelle génération  $P'$  en prenant les  $n$  meilleurs individus de la population  $P$  après l'avoir triée de manière décroissante selon la fitness de ses individus. Il est inutile de préciser que cette méthode est encore pire que celle de la loterie biaisée dans le sens où elle amènera à une convergence prématurée encore plus rapidement et surtout de manière encore plus sûre que la méthode de sélection de la loterie biaisée. En effet, la pression de la sélection est trop forte, la variance nulle et la diversité inexistante, du moins le peu de diversité qu'il pourrait y avoir ne résultera pas de la sélection mais plutôt du croisement et des mutations.

### 5.2.3 La sélection par tournois

Le principe de cette méthode est le suivant : on effectue un tirage avec remise de deux individus d'une population  $P$ , et on les fait "combattre". Celui qui a la fitness la plus élevée l'emporte avec une probabilité  $p$  comprise entre 0.5 et 1. On répète ce processus  $n$  fois de manière à obtenir les  $n$  individus de la nouvelle population  $P'$  qui serviront de parents. La variance de cette méthode est élevée et le fait d'augmenter ou de diminuer la valeur de  $p$  permet respectivement de diminuer ou d'augmenter la pression de la sélection [25].

### 5.2.4 La sélection universelle stochastique (SUS)

Pour comprendre cette méthode de sélection, imaginons une bande de longueur  $L$  sur laquelle les individus de la population occupent un espace proportionnel à leur performance et sont classés en ordre de performances décroissantes. Un exemple est représenté à la figure 2.5 pour une population de dix individus dans laquelle sept d'entre eux doivent être sélectionnés.



**Figure 2.5:** Sélection universelle stochastique.

Les individus sont sélectionnés par des curseurs : lorsque le curseur se retrouve dans la case d'un individu, ce dernier est sélectionné. Au départ, le premier curseur est positionné aléatoirement dans l'intervalle  $[0, L/N_{sel}]$ , où  $N_{sel}$  est le nombre d'individus que l'on veut sélectionner pour la reproduction ( $N_{sel} = 7$  dans l'exemple). Par la suite, les autres pointeurs sont également espacés de  $L/N_{sel}$ . Dans l'exemple, les individus 0, 1, 2, 3, 5 et 7 sont sélectionnés (en gris à la figure 2.5). Comme on peut le voir, il est possible qu'un même individu soit sélectionné plus d'une fois. Cependant, la SUS réduit ce phénomène comparativement à d'autres méthodes de sélection, comme la roulette par exemple. En effet, avec la méthode de la roulette, le même individu aurait pu être sélectionné jusqu'à sept fois dans l'exemple. Avec la SUS, les individus les plus performants ont une plus grande probabilité d'être sélectionnés, mais des designs peu performants peuvent aussi l'être. Cependant, de la même façon que la sélection naturelle sur Terre, les mauvais designs sélectionnés par hasard finiront forcément par s'éteindre dans les générations futures. [38]

### 5.3 Le croisement

Dans cette étape, la population qualifiée passera par le croisement ou de nouveaux individus ou enfants seront obtenus des parents. Dans ce là les individus seront choisis au hasard pour former des couples. Les parents de chaque couple doivent échanger des parties de leurs chromosomes pour former de nouveaux individus (enfants). Cette étape est contrôlée par le type de croisement choisie et la probabilité de croisement.



### 5.3.1 Croisement multipoints

Pour ce type de croisement,  $p$  positions de croisement, avec  $p \in [1, 2, \dots, l-1]$  où  $l$  représente la taille du chromosome, sont choisies aléatoirement pour chaque couple. Ensuite, les gènes entre deux points de croisement successifs sont échangés entre deux parents produisant deux nouveaux individus appelés enfants. Les valeurs typiques de  $p$  sont : 1 (croisement à 1 point) et 2 (croisement à 2 points) [31].

#### ➤ Croisement à un point

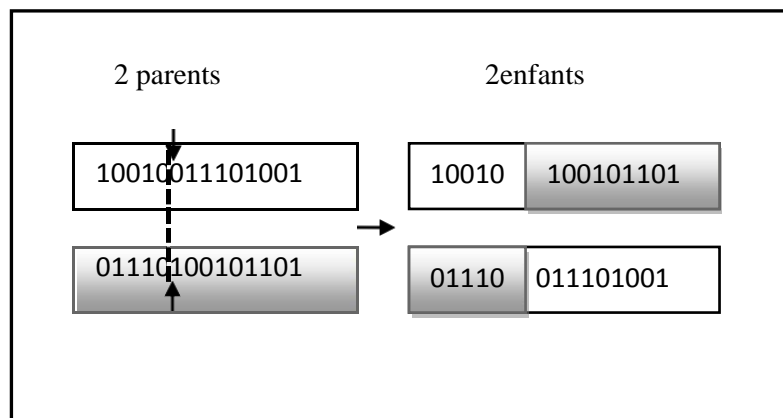
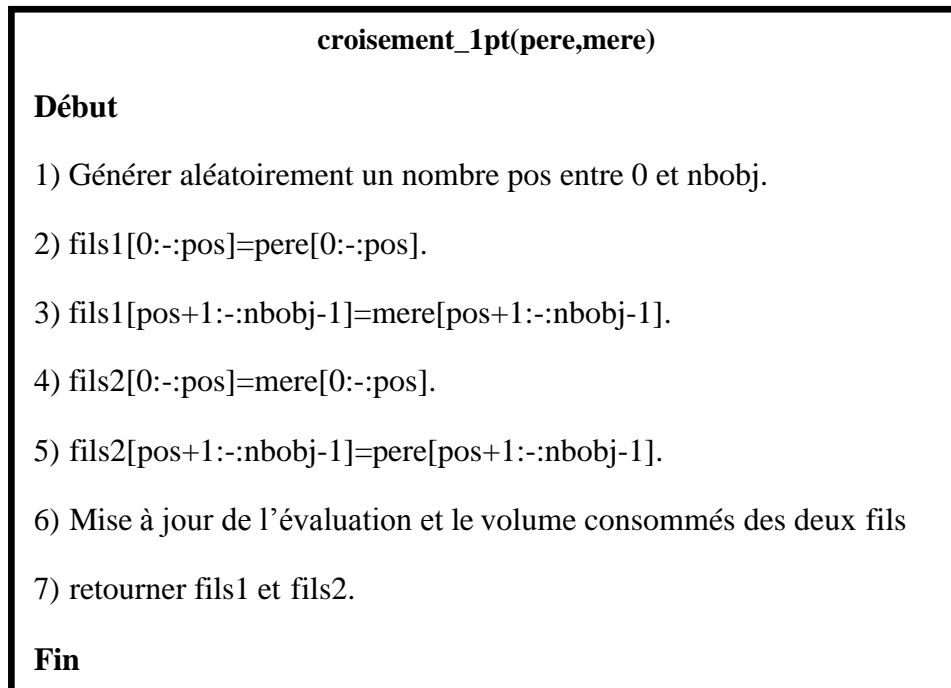


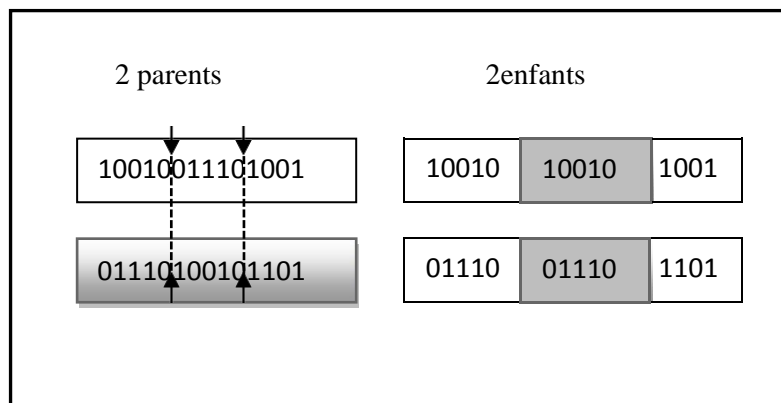
Figure 2.6: Croisement avec un point [31].

❖ L'algorithme suivant formalise le principe du croisement à un point .



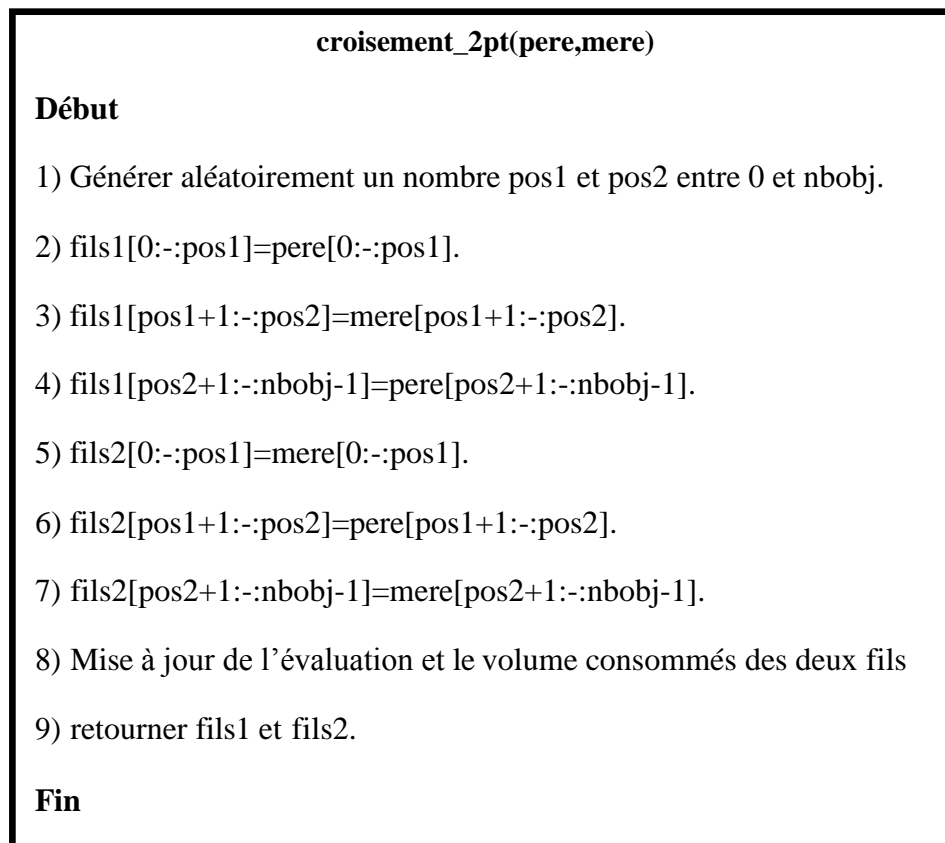
Algorithme 2.1: Croisement à un point [39].

## ➤ Croisement à deux points



**Figure 2.7:** Croisement avec 2 points) [31].

❖ L'algorithme 2.2 formalise le principe du croisement à deux points :

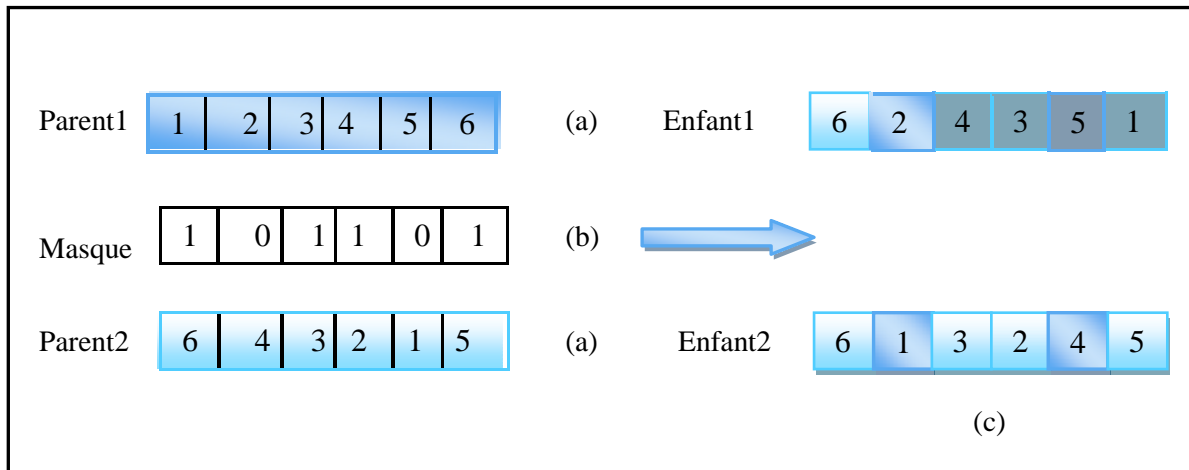


**Algorithme 2.2:** Croisement à deux points [39].

### 5.3.2 Croisement uniforme

Le croisement uniforme de permutations tend à faire hériter un enfant d'une combinaison des ordres existants dans deux parents. Le croisement se déroule en trois étapes :

- ❖ Un masque binaire est engendré aléatoirement, figure(b) ;
- ❖ Deux parents sont appariés. les (0) du masque binaire définissent les positions préservées dans la séquence du parent 1, et les (1) du masque binaire définissent les positions préservées dans la séquence du parent 2, figure(a) ;
- ❖ Pour obtenir l'enfant 1, les éléments non préservés du parent 1 sont permutés de façon à respecter l'ordre qu'ils ont dans le parent 2, figure(c) [42].



**Figure2. 8:**Illustration du croisement uniforme[42].

## 5.4 Mutation

Le rôle de cet opérateur est de modifier aléatoirement, avec une certaine probabilité, la valeur d'un composant de l'individu. Cette modification de la valeur du chromosome produit un nouvel individu.

Une mutation consiste simplement en l'inversion d'un bit (ou de plusieurs bits, mais vu la probabilité de mutation c'est extrêmement rare) se trouvant en un locus bien particulier et lui aussi déterminé de manière aléatoire. L'opérateur de mutation modifie, donc, de manière complètement aléatoire les caractéristiques d'une solution, ce qui permet d'introduire et de maintenir la diversité au sein de notre population de solutions. Cet opérateur joue le rôle d'un "élément perturbateur", il introduit du "bruit" au sein de la population [25].

Les plus connus des types de mutation sont ceux présentés ci-dessous :

### 5.4.1 Insertion

L'opérateur d'insertion choisit un gène aléatoirement et l'insère dans une autre position du chromosome.

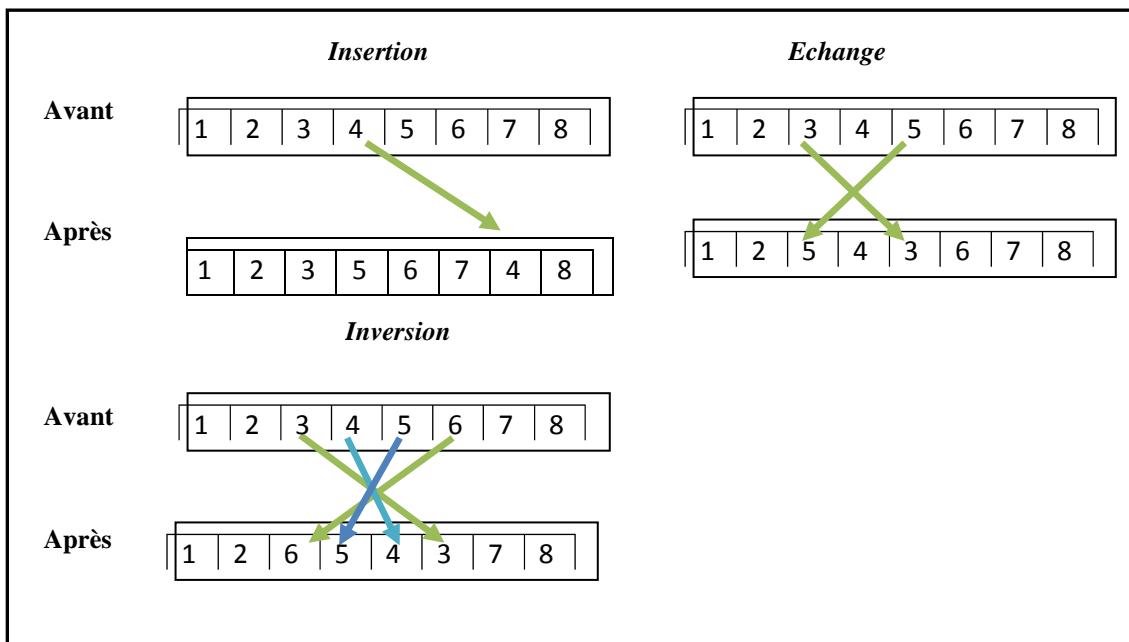
### 5.4.2 Echange (1-Opte ou Swap mutation)

L'opérateur d'échange consiste à prendre, au hasard, deux gènes du chromosome et à les permuter.

### 5.4.3 Inversion

L'opérateur d'inversion inverse l'ordre de gènes entre deux points de coupure choisis aléatoirement.

➤ La figure 2.9, donne des exemples les trois mutations précédentes.



**Figure2.9:**Opérateur d'insertion, échange et inversion.

### 5.4.4 $\lambda$ -Opt

L'opérateur  $\lambda$ -Opt enlève  $\lambda$  arcs (avec  $\lambda = 2, 3, 4, \dots, n$ ) pour remettre les chaînes associées dans la meilleure combinaison possible.

### 5.4.5 Or-Opt

L'opérateur Or-Opt opère de la même manière que la mutation 1-Opt, mais les tailles des suites choisies peuvent varier entre 0 et 3 (figure2.10). Cet opérateur échange deux suites ou il déplace une suite vers une autre position si la taille de l'une des suites est égale à zéro.

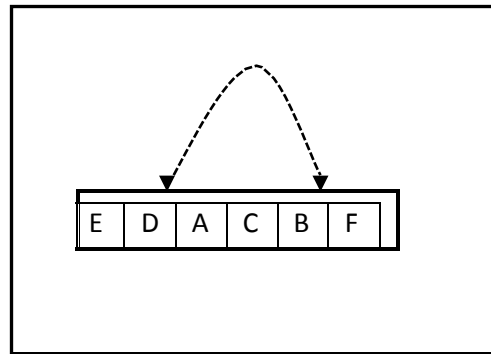


Figure2.10:Or-opt[.

#### 5.4.6 Mélange (Scramble Mutation ou SM)

Cette mutation fut proposée par (Syswerda, 1991). Elle permute aléatoirement les gènes entre deux points de coupure choisis au hasard comme dans la figure (2.11).

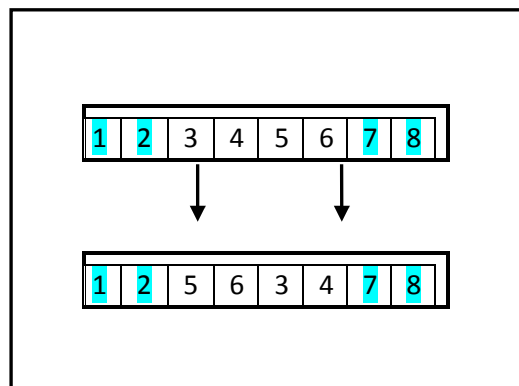


Figure2.11:Exemple de scramble Mutation.

Noter que les mutations (1-Opt,  $\lambda$ -Opt et Or-opt) peuvent être trouvées dans la littérature sous le nom de mutation échange (Exchange Mutation ou EM).

#### 5.5 Critère d'arrêt

Généralement, le cycle de génération et remplacement est répété jusqu'à ce qu'un critère d'arrêt soit satisfait. Ce critère peut être notamment un nombre fixe d'itérations (générations), un temps maximal de calcul, ou/et une solution satisfaisante trouvée. L'algorithme évolutionnaire retourne, alors, la (ou les) meilleure(s) solution(s) qu'il a identifiée(s) de génération en génération [42].

## 6. Avantages des AGs

Les algorithmes génétiques ont plusieurs avantages citons les suivants :

- ❖ Ils sont adaptables à plusieurs types de problèmes.
- ❖ Les AGs présentent une grande robustesse, c'est-à-dire une grande capacité à trouver les optimums globaux des problèmes d'optimisation.
- ❖ Facile à implémenter.
- ❖ Enfin, Les AGs sont aisément parallélisables [45].

## 7. Conclusion

Dans ce chapitre nous avons présenté une vue générale sur les algorithmes génétiques, leurs paramètres leurs principaux opérateurs et leurs types, ainsi que le principe de base d'un AG standard.

## CHAPITRE 3

# CONCEPTION

### 1. Introduction

**D**ans le premier chapitre, nous avons présenté la congestion routière et la gestion du trafic routier en général et les systèmes intelligents pour la régulation ainsi que les outils théoriques qui sont utilisés pour la modélisation du trafic routier. Nous avons, en particulier, mis l'accent sur la théorie des algorithmes génétiques que nous avons utilisée pour modéliser notre système.

Ainsi, On a détaillé le principe des algorithmes génétiques qui on peut l'utilisé pour développé un système intelligent des feux de circulation peut être résolu par une procédure d'optimisation par AG qui à partir d'un ensemble de solutions initiales, ou population de N individus, elle consiste à faire évoluer cette population en utilisant des opérateurs de sélection, de croisement et de mutation. A chaque itération de l'algorithme, une nouvelle population de solutions ou d'individus est générée. Tout d'abord, un ensemble d'individus est sélectionné pour générer la population suivante. Ces individus sont ensuite croisés pour créer de nouveaux individus et compléter la nouvelle population. Certains des individus peuvent subir une mutation. et l'algorithme rester en exécution .

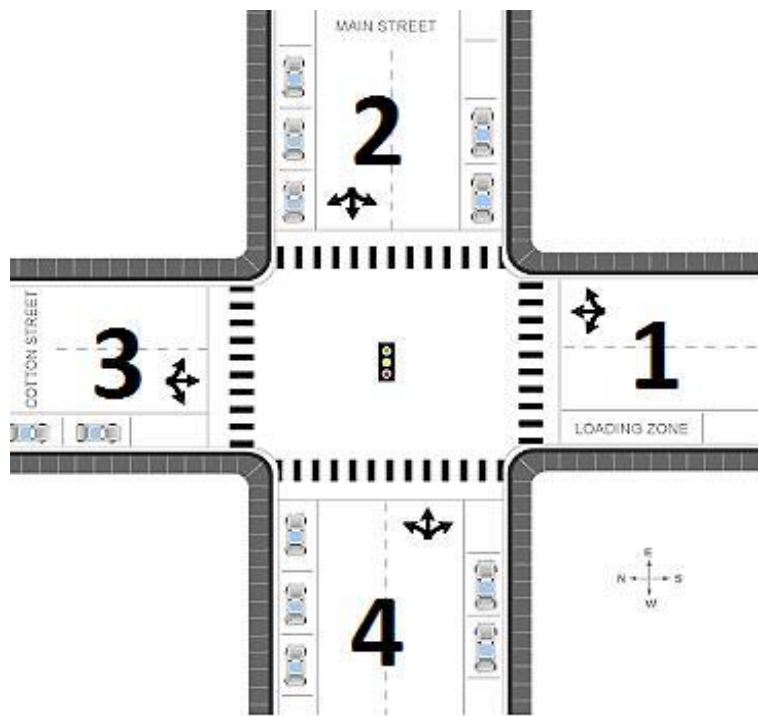
La gestion des feux de circulation, et la prise de décisions pour chacun, est extrêmement difficile et complexe. Pour simuler le monde réel, plusieurs hypothèses ont dû être formulées afin de tenir compte de toutes les situations possibles qui peuvent se produire normalement ou anormalement, comme les accidents de voiture, les événements spéciaux (événements sportifs, festivals, etc.) et même les situations d'urgence. Mais ici, nous simulons notre système pour des conditions normales.

Donc le but c'est que comment on peut avoir une meilleur distribution du feux vert t sur les route et ne le Gasperi pas on parallèle avec le et maximisation du passage des véhicule tout ca le système prend ces décisions en temps réel. Cela le but de notre travail, où nous présentons à travers ce chapitre suivant le mode de codage adopté et les déférent étapes a suivre.

## 2. Formalisation du problème de feux de circulation

Le système est basé sur un algorithme génétique qui reçoit les entrées du système de détection d'image vidéo qui prendra une décision et déterminera le temps de lumière verts pour minimiser les congestions et le flux d'embouteillages.

La détection et le comptage des véhicules sont importants dans le calcul de la congestion routière, ce qui représente les intrants de notre système. Les objets ici sont définis comme des véhicules se déplaçant ou s'arrêtant sur les routes. Il y a des véhicules de différentes tailles qui empruntent la route, dont certains sont des voitures et des autobus.. Le nombre de véhicules de chaque côté de l'intersection est compté lorsque le feu de circulation est rouge. La figure 1 montre un exemple de l'intersection entre deux routes principales.

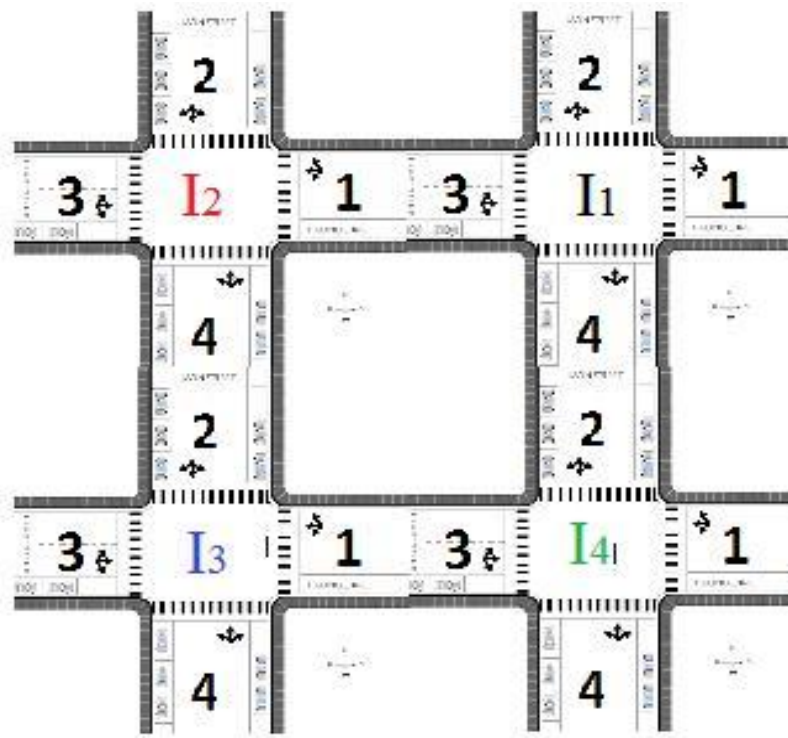


**Figure 3.1** Intersection entre deux routes principales



Pour des fins de comparaison abordées sur le prochain chapitre, nous optons pour les mêmes hypothèses formulées dans [45], à savoir :

- Dans la figure 3.2, chacune des intersections I1, I2, I3 et I4, est contrôlée par un feu de circulation intelligent.
- On suppose que les intersections sont relativement « occupées » et sous-saturées, avec des variations importantes de la demande dans toutes les approches.
- L'intersection est supposée être l'étiquette de quatre routes en suivant les numéros 1, 2, 3 et 4.
- Tous les véhicules roulent à la même vitesse de 40 km/h.
- La distance entre chaque intersection est de 200 m.



**Figure 3.2** La proposition de quatre intersections

### 3. Feu de circulation génétique

#### 3.1 le codage

L'étape de codage est l'une des étapes les plus importantes de l'algorithme génétique d'un chromosome. Une solution candidate (chromosome) dans notre cas est un vecteur de 16 nombres aléatoires (gènes). Chaque gène correspondant à un temps de feu vert est constitué d'un nombre aléatoire, entre 15 et 35 secondes.

Réellement, c'est le codage binaire qui est considéré pour permettre la génération (par application des opérateurs de reproduction) de nouvelles valeurs de gènes (temps verts) permettant de s'adapter aux changements réels de circulation sur la route.

$Tv_{11}$	$Tv_{12}$	$Tv_{13}$	...	$Tv_{42}$	$Tv_{43}$	$Tv_{44}$
-----------	-----------	-----------	-----	-----------	-----------	-----------

**Figure 3.3** Codage adapté

Tels que :

- $tv_{ij}$  : est le temps vert du feu de l'intersection  $i$  dans la route  $j$

- $15 \text{ secondes} < tv_{ij} < 35 \text{ secondes}$  ,  $tv_{ij}$  peut prendre aussi la valeur 0 (feu rouge allumé).

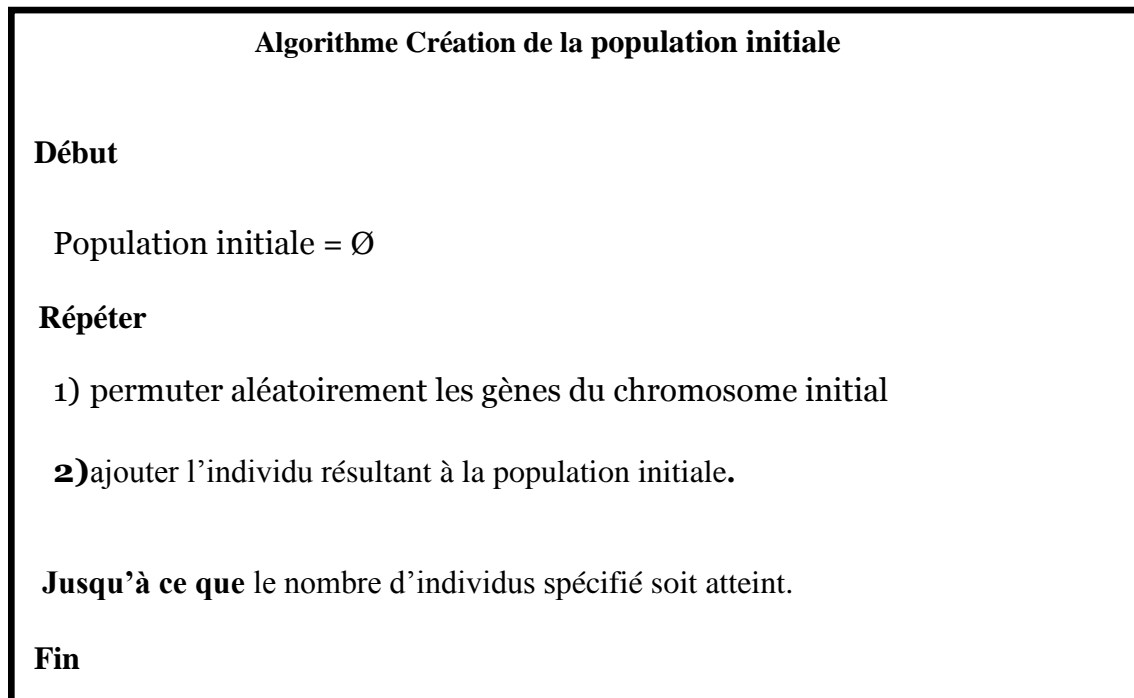
#### 3.2. Création des populations initiales

Après l'étape de codage, la création de population est l'étape qui suit, ou de nouveaux

individus seront créés pour former la population initiale qui sera la première à passer par la reproduction et l'évaluation, pour aboutir à la sélection en dernier lieu.

Dans notre cas, la population initiale sera composée d'un certain nombre d'individus créés chacun en permutant aléatoirement les gènes du chromosome initial codant une attribution initiale des temps de feux verts des différentes routes des quatre intersections suivant le nombre de véhicules initial (entrée du système).

L'algorithme suivant résume les étapes de la création de la population initiale



**Algorithme 3.1** Création de la population initiale

### 3.3 Opérateurs de reproduction

#### 3.3.1 Croisement

Le croisement permet d'obtenir de nouveaux individus parmi les individus existants dans la population, ce qui amélioré la qualité de la population. Cette étape sera effectuée en choisissant deux individus de la population et en appliquant le croisement sur eux, le résultat sera un ou deux nouveaux individus (le résultat est un individus si on obtiens deux individus identiques) .

L'algorithme suivant illustre les étapes du croisement.

### Algorithme de croisement

#### Début

- 1- Sélection de deux individus au hasard.
- 2- Génération de deux points de croisement au hasard.
- 3- Application de l'opérateur de croisement choisi.
- 4- Ajoute des enfants à la population.

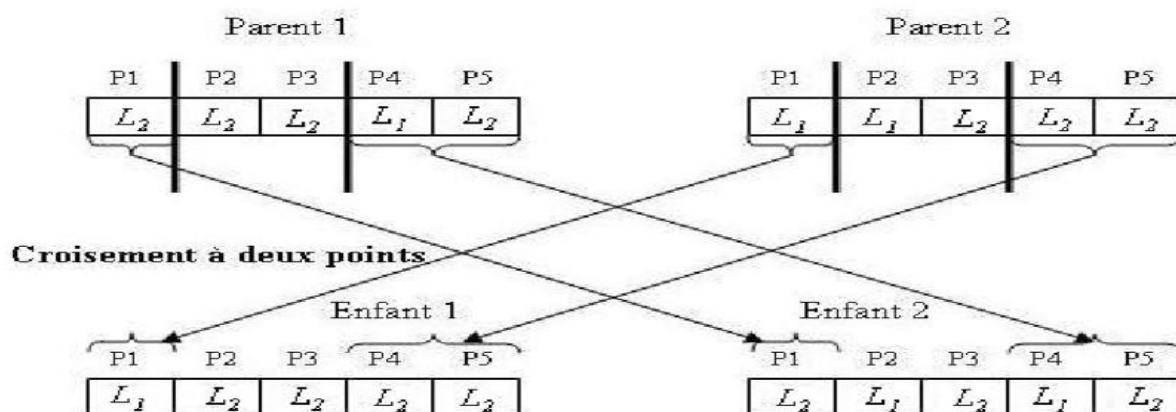
#### Fin

### Algorithme 3.2 : Algorithme de croisement.

Pour notre conception, l'opérateur de croisement choisis est le croisement à deux points dont les étapes se résument en :

- 1- choisir dans les deux parents une sous-séquence interne, comprise entre deux points de coupure tirés aléatoirement.
- 2- Recopier la première séquence du Parent1 dans le descendant Enfant1 aux mêmes positions.
- 3- Retirer de la séquence interne chromosome Parent2 et compléter la séquence interne du Enfant1 .
- 4- Compléter le reste du chromosome Enfant1 la deuxième séquence du Parent1 en faire même chose avec Enfant2.

La figure suivante donne un exemple applicatif du croisement à deux points.



**FIGURE 3.4** Exemple du croisement à deux points

### 3.3.2 Mutation

L'opérateur de mutation manipule les gènes d'un individu choisi au hasard. Dans ce cas, la mutation sera appliquée à l'individu en effectuant une permutation entre deux gènes de l'individu. L'individu résultant sera ajouté à la population.

L'algorithme suivant montre les étapes de l'opérateur de mutation.

#### Algorithm 3.3 Mutation

##### Début

- 1) sélectionner un individu aléatoirement
- 2) choisir deux positions  $i$  et  $j$  tel que  $0 \leq i, j \leq \text{taille de l'individu}$  et  $i \neq j$ .
- 3) permuter entre le gène  $i$  et le gène  $j$  de l'individu pour produire un individu fils.
- 4) ajouter le fils à la population.

##### Fin

#### Algorithme 3.3 : Algorithme de Mutation

### 3.4. L'évaluation des individus

La fonction d'évaluation quantifie la qualité de chaque chromosome par rapport au problème. Les chromosomes ayant une bonne qualité ont plus de chance d'être sélectionnés pour la reproduction, et donc plus de chance pour survivre d'une génération à une autre. La fonction d'adaptation produit la pression qui permet de faire évoluer la population de l'algorithme évolutionnaire vers les individus de meilleure qualité. En clair, le choix de la fonction d'évaluation va fortement influencer sur le succès de l'algorithme.

- ✓ La fonction d'évaluation que nous avons choisie pour associer des valeurs d'adaptation à chaque individu est la suivante :

$$F(i) = \text{Max} 1 / \sum_{i=1}^4 w_i$$

Où :

- **I** est le  $i^{\text{ème}}$  individu d'une certaine population.
- **W** est le Poids de trafic est calculé pour chaque intersection

$$w_i = \sum_{m=1}^4 C'_{im}$$

- ✓ Pour calculé le changement de nombre de véhicule nous optons pour les mêmes hypothèses formulées dans [45], à savoir que chaque véhicule a besoin en moyenne de 3 secondes pour quitter l'emplacement actuel. Donc, le nombre de véhicules après consommation du feu vert, noté  $C'_{im}$ , est donné par :

$$C'_{im} = C_{im} - tv/3$$

Où :

- **C<sub>im</sub>** est le nombre des véhicules dans l'intersection ( i ) sur la route ( m ) au moment où c'est le feu rouge qui est allumé.
- **tv** est le tempe du feu vert donné à une route .

### 3.5. La sélection des individus

Après l'évaluation, les individus sont maintenant prêts passer par l'étape de sélection Le rôle de la sélection est de distinguer entre les individus sur la base de leur qualité, en particulier, pour permettre aux meilleurs individus de devenir parents dans la génération suivante. Ainsi, elle est responsable sur le fait de pousser l'amélioration de la qualité.

Dans notre cas l'opérateur utilisé est une sélection de type roulette avec la possibilité de sélectionner plusieurs fois le même individu. Ainsi, les meilleurs individus ont plus de chance d'être sélectionnés par rapport au moins bons individus.

Avec cette technique, les parents sont sélectionnés en fonction de leur performance, où les individus ayant une bonne évaluation, auront plus de chances d'être sélectionnés. Son principe ressemble à celui de la roulette de casino, du fait qu'il revient à imaginer une sorte de roulette de casino sur laquelle sont placés tous les chromosomes de la population suivant leurs valeurs d'adaptation.

Ensuite, la bille est lancée et s'arrête sur un chromosome. Les points suivants représentent les étapes de l'algorithme de sélection qu'on a appliqué :

- 1) On calcul la somme  $S1$  de toutes les valeurs d'évaluation des individus d'une population.
- 2) On génère un nombre  $r$  entre  $0$  et  $S1$ .
- 3) On calcule ensuite une somme  $S2$  des évaluations en s'arrêtant dès que  $r$  est dépassé
- 4) Le dernier chromosome dont la fonction d'évaluation vient d'être ajoutée est sélectionné

#### **Algorithme Sélection par roulette**

Population  $i$  : population à l'itération  $i$ . Liste\_sélect : la liste des individus sélectionnés.  $N$  : nombre d'individus à sélectionner.

#### **Début**

*Pour*  $i=1, N$  *faire* // sélectionner  $N$  individus

- 1) Générer un nombre aléatoire  $U$  ;
- 2) Sélectionner un  $individu_j$ , tel que :  $Eval (individu_{j-1}) \leq U < Eval (individu_j)$
- 3) Ajouter l' $individu_j$  à list\_sélect;

*Fin pour*

#### **Fin**

**Algorithme 3.4** Algorithme de la sélection par roulette.

### 3.6. Critère d'arrêt

Après l'exécution des étapes précédentes (croisement, mutation, évaluation, sélection) pour un nombre spécifique d'itération représentant le nombre de génération produites, l'exécution sera terminée lorsque le nombre d'itération atteindra le nombre spécifié.

## 4. Points forts des algorithmes génétiques

Les algorithmes génétiques sont utilisés pour calculer des solutions optimales dans différents problèmes de différents domaines. Dans notre cas d'étude, les points suivants représentent les points forts essentiels participant à la conception d'un bon système intelligent de gestion des feux de circulation.

- ❖ Tout d'abord, le concept de l'algorithme génétique est très simple et peut être facilement compris et mis en œuvre.
- ❖ Les algorithmes génétiques lancent la recherche d'une solution optimale parmi un groupe de solutions et non une solution unique.
- ❖ Le codage binaire utilisé permet de générer des nouvelles valeurs de temps de feux verts pouvant apparaître à chaque exécution; permettant, ainsi, de s'adapter aux nouvelles situations et exigences de la route.
- ❖ Les algorithmes génétiques utilisent des règles probabilistes pour trouver la solution optimale et non pas des règles déterministes, qui offrent davantage de diversité.

## 5. Conclusion

Dans ce chapitre, les détails de la structure de ce travail ont été introduits et expliqués. nous avons exposé en détail les différentes étapes qui constituent la structure générale du processus développé : codage, création de la population initiale, opération de croisement et de mutation, méthode de sélection et le test d'arrêt.



## CHAPITRE 4

# EXPERIMENTATION ET RESULTATS

### 1. Introduction

**N**ous avons pu profiter dans les chapitres précédents d'une présentation des modèles, principes et méthodes caractérisant le processus de résolution du problème traité qui est celui de gérer un feu de circulation intelligente à base des Algorithmes génétiques. Selon cette approche, nous avons présenté dans le précédent chapitre, le modèle (l'algorithme) génétique de base que nous cherchons à optimiser le temps perdu pour le véhicule en proposant un schéma de. En effet, un algorithmes ont été proposés dont une description conceptuelle a été présentée dans le chapitre précédent.

A travers le présent chapitre, nous présentons le détail de développement de l'algorithmes de génération des feux génétiques vus dans le troisième chapitre. Nous décrivons l'implémentation de nos algorithmes en justifiant certains choix techniques tels que l'environnement de programmation utilisé. Une série de tests servant à régler les paramètres des algorithmes développés sera bien explicitée, suivie d'une énumération de résultats expérimentaux obtenus par application de l'approche de simulation proposée sur différentes cas. Ainsi ce chapitre sera subdivisé en cinq sections.

- Dans la première section nous nous intéressons à la présentation de l'environnement de développement
- La deuxième section sera réservée pour présenter la structure du programme et à expliquer le rôle des classes implémentant les différentes phases du processus de calcul génétique.
- La troisième section indique le choix des paramètres génétiques de nos algorithmes.
- A travers La quatrième section, nous rapportons les résultats expérimentaux obtenus par notre modèle.
- La cinquième section est dédiée à la présentation des fenêtres de l'application.

## 2. L'environnement de programmation JAVA

Il existe plusieurs langages de programmation supportant la simulation . Nous avons choisi le java pour les raisons suivantes :

- ❖ Java est portable : Java est indépendant de toute plate-forme et il n'y a pas de compilation spécifique pour chaque plate forme. Le code reste indépendant de la machine sur laquelle il s'exécute. Il est possible d'exécuter des programmes Java sur tous les environnements qui possèdent une Java Virtual Machine.
- ❖ Java est simple : il n'utilise pas la notion des pointeurs (pour éviter les incidents en manipulant directement la mémoire on interfère par le temps de calcul), l'héritage multiple et la surcharge des opérateurs,
- ❖ Java est un langage de programmation multi-thread : qui signifie que nous pouvons développer un programme multi-thread avec Java. Un programme multi-thread contient deux parties ou plus qui peuvent s'exécuter simultanément et chaque partie peut gérer une tâche différente tout en optimisant les ressources disponibles. C'est principalement à cause de cette fonctionnalité que nous avons choisi JAVA puisqu'elle convient parfaitement avec nos besoins de calcul parallèle.

## 3. La structure du programme

Dans cette section nous présentons une description de la structure de l'application qui englobe l'ensemble des classes suivantes :

- **Classe Panel** : Cette classe contient les méthodes suivantes :
  - **La méthode `initIU()`** : Cette méthode permet de dessiner les quatre intersections.
  - **La méthode `creerFeu()`** : Cette méthode est responsable de la création des feux de circulation sur les routes (placement des feux sur les routes).
  - **La méthode `lancerSimulation()`** : Cette méthode permet de lancer la simulation (lancer des véhicules aléatoirement sur les intersections).
- **Classe Car** : Cette classe regroupe les paramètres des véhicules : leurs positions et leurs déplacements sur les différentes intersections.
- **Classe Case** : Cette classe définit les positions des différents composants sur les routes. Donc, elle a comme rôle de connaître l'emplacement des véhicules et des feux de circulation.

- **Classe Chromosome** : Cette classe contient une méthode qui crée des populations de manière aléatoire à partir du chromosome initial en plus d'une autre méthode quantifiant la qualité de chacun des individus (méthode d'évaluation).
- **Classe OperateursGenitique** Cette classe contient les méthodes suivantes :
  - **La méthode crossover** : cette méthode permet de choisir deux individus aléatoirement et leur applique l'opération de croisement. Le résultat sera deux nouveaux individus qui seront ajoutés à la population.
  - **La méthode mutation** : Ici , un individu est choisi au hasard ou une opération de permutation sera appliquée entre deux gènes choisis aléatoirement. Le résultat est un nouvel individu qui sera rajouté à la population.
  - **La méthode selection** : cette méthode implémente l'application de la roulette en utilisant les scores d'évaluation. Elle maintient les individus qui formeront la nouvelle population.
- **Classe GererFeu** : Cette classe donne la permission de lancer les feux verts.

#### 4. Paramètres génétiques et illustrations

A travers cette section, nous allons présenter la démarche suivie pour régler les paramètres des Algorithmes et illustré d'implémentation des algorithmes génétiques.

##### 4.1 paramètres génétiques

Pour les paramètres génétiques qui incluent la probabilité de croisement, la probabilité de mutation, le nombre de générations et la taille des populations, le tableau suivant résume leurs valeurs.

Probabilité du croisement	Probabilité de la mutation	Le nombre de Génération	La taille de la population (nombre d'individus)
0.6	0.007	100	10

**Tableau 4.1** Paramètre génétique adoptés.

4.2 Illustration

4.2.1. Codage des chromosomes

Comme c'était présenté dans le chapitre précédent, dans notre cas, le chromosome est formé d'entiers représentant le temps vert pour chaque route des quatre intersections

Dans la figure 4.1, un exemple d'initialisation de chromosome est illustré. Le vecteur correspondant énumère le temps des feux verts des différentes routes des quatre intersections.

{ les feux de intersection I1 }				{ les feux de intersection I2 }				{ les feux de intersection I3 }				{ les feux de intersection I4 }			
<u>0</u>	17	<u>0</u>	17	25	<u>0</u>	25	0	30	<u>0</u>	30	<u>0</u>	<u>0</u>	20	0	20

**Figure 4.1** Exemple d'initialisation du temps des feux verts.

4.2.2. Génération de la population initiale

La figure 4.2 donne un exemple de deux nouveaux chromosomes générés à partir du chromosome initial présenté par la figure 4.1 en permutant des gènes choisis aléatoirement.

19	0	19	0	0	31	0	31	17	0	17	0	25	0	25	0
----	---	----	---	---	----	---	----	----	---	----	---	----	---	----	---

(a)

0	28	0	28	0	34	0	34	0	15	0	15	0	21	0	21
---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----

(b)

**Figure 4.2** Exemple de chromosomes générés à partir du chromosome initial illustré par la figure 4.1, (a) : chromosome 1, (b) : chromosome 2.

Grâce au codage adopté qui est le codage binaire des valeurs entières représentant le temps des feux verts, nous remarquons que les deux chromosomes (voir figure 4.2) générés à partir du chromosome initial donné par la figure 4.1, comportent de nouvelles caractéristiques (nouvelles valeurs de gènes, donc nouvelles valeurs de temps des feux verts) par rapport au chromosome initial.

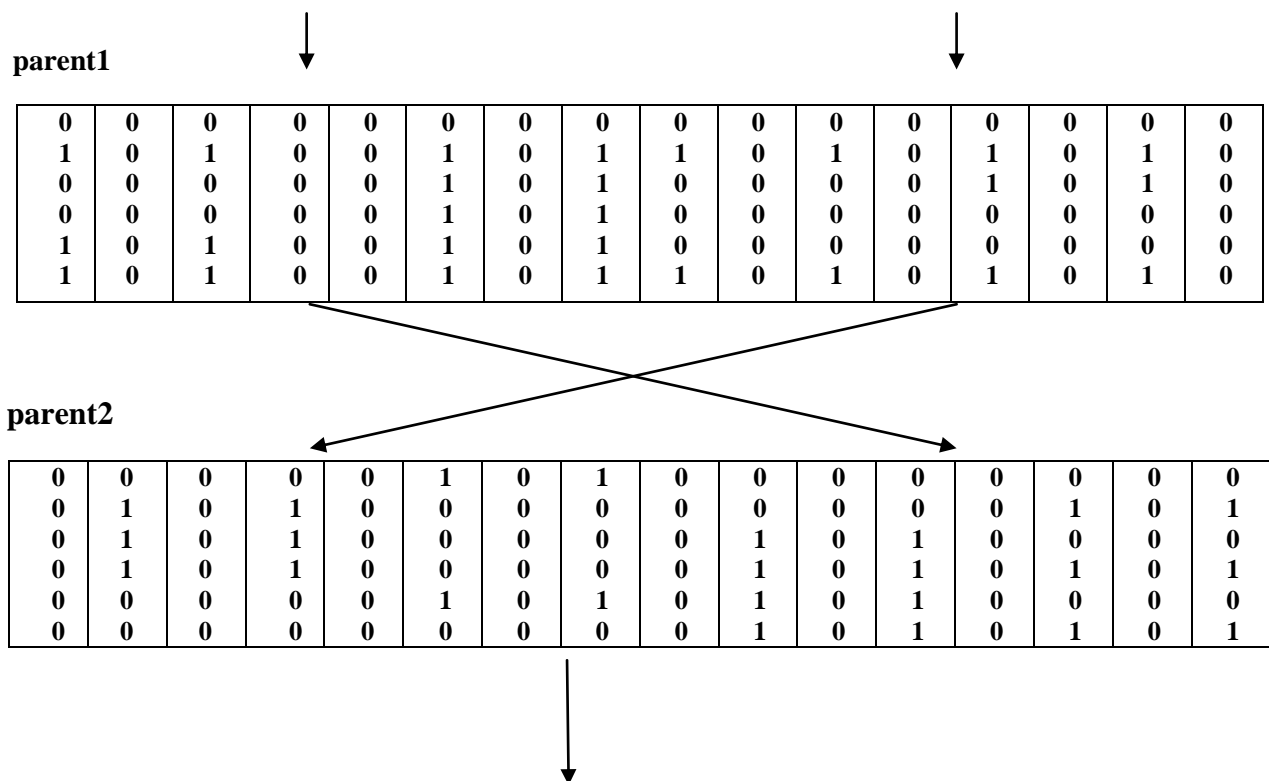
Par exemple, les valeurs 19, 31, 25 sont de nouvelles valeurs apparaissant dans le premier chromosome et ne figurant pas dans le chromosome initial. Toutefois, d'autres valeurs ont disparu et qui sont : 30 et 20.

De même, pour le deuxième chromosome de nouvelles valeurs de gènes apparaissent et d'autres disparaissent.

### 4.2.3. Opérateurs de reproduction

#### 4.2.3.1. Croisement

Supposant l'application d'une opération de croisement entre les deux chromosomes donnés par la figure 4.2, les deux enfants générés sont donnés par la figure 4.3.



enfant1

0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
1	0	1	0	0	1	0	1	0	0	0	0	1	0	1	0
0	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0
1	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	1	0	1	0	1	0	1	0	0	0	0



enfant1

23	0	23	0	0	29	0	29	0	33	0	33	24	0	24	0
----	---	----	---	---	----	---	----	---	----	---	----	----	---	----	---

**Figure 4.3** Exemple d'enfants calculés par croisement entre les parents de la figure 4.2.

**4.2.3.2.Mutation**

La figure 4.4 présente un enfant généré par application de l'opération de mutation sur le chromosome 1 donné par la figure 4.2.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	1	0	1	0	1	0	1	0	0	1	0	1
1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0
1	0	1	0	1	0	1	0	1	0	1	0	0	1	0	1

31	0	31	0	19	0	19	0	17	0	17	0	0	25	0	25
----	---	----	---	----	---	----	---	----	---	----	---	---	----	---	----

**Figure 4.4** Exemple d'un enfant généré par mutation appliquée sur le chromosome 1 de la figure 4.2.

## 5. Présentation des interfaces

Le but de notre application est de simuler un système de feux de circulation à l'aide d'un algorithme génétique. A travers notre conception de fenêtre répandue aux différents cotés du travail et ses différentes fonctionnalités, nous cherchions à ce que l'application soit d'un usage simple. Dans cette section, nous décrivons les différentes fenêtres constituant l'application.

### 5.1 La fenêtre d'accueil

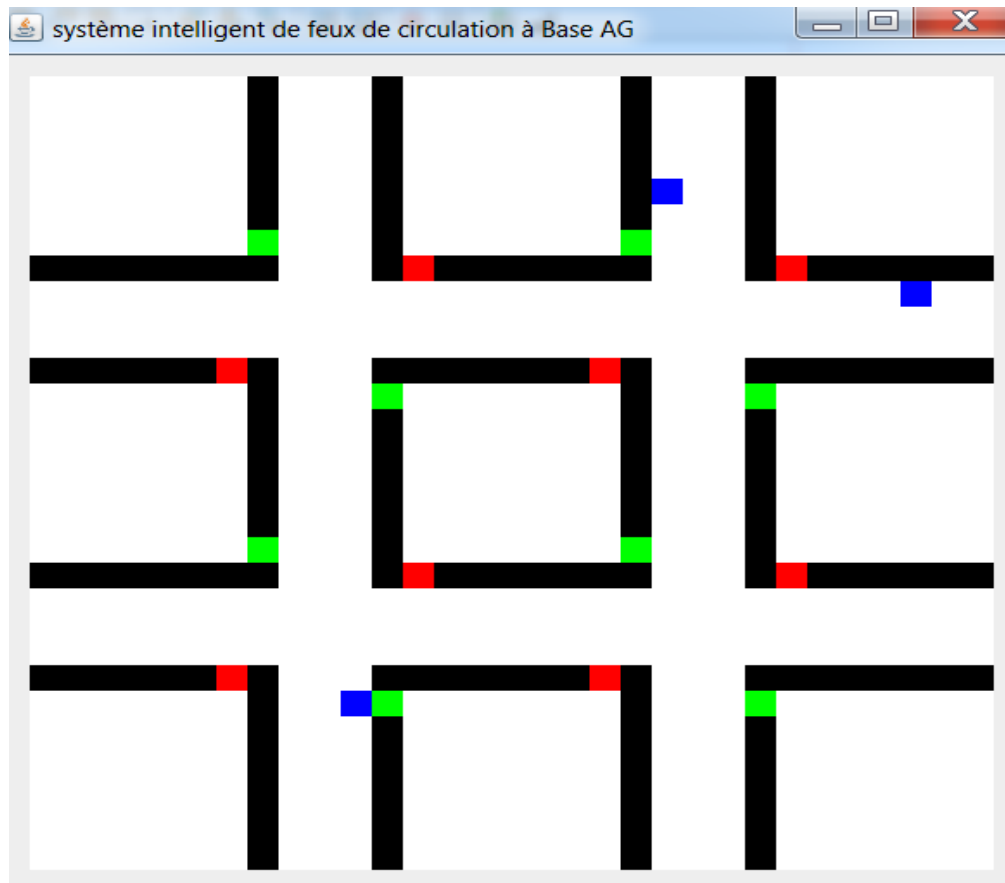
Lorsque l'utilisateur lance l'application, la première qui s'affiche est la fenêtre d'accueil (figure 4.5). Cette fenêtre lui donne la possibilité d'accéder aux simulations du système sur le bouton Simuler, ou de quitter l'application en utilisant le bouton Quitter.



FIGURE 4.5 La fenêtre d'accueil

### 5.2 La fenêtre de simulation

Cette fenêtre qui est représentée dans la figure 4.6 est celle qui va implémenter le système que nous avons proposé dans le chapitre précédent. Cette fenêtre contient l'illustration d'une intersection, au bord de chaque route on trouve un feu de circulation qui prend la couleur verte et rouge, les véhicules circulent aléatoirement sur les routes, ils sont représentés par le carré bleu qui existe dans la fenêtre.



**FIGURE 4.6** La fenêtre de simulation

## 6. Résultats expérimentaux et discussion

Nous rapportons, dans cette section, des exemples d'application de notre système sur différents états de la route et on présentera les décisions prises.

### 6.1. Résultats expérimentaux

#### 6.1.1 Exemple 1

Soit un état de la route représenté par la figure 4.7, d'après lequel un problème de congestion apparaît principalement sur la route 1 de l'intersection I1, et sur d'autres routes à un degré moindre (route 3 de I2, route 3 de I3 et route 1 de I4).



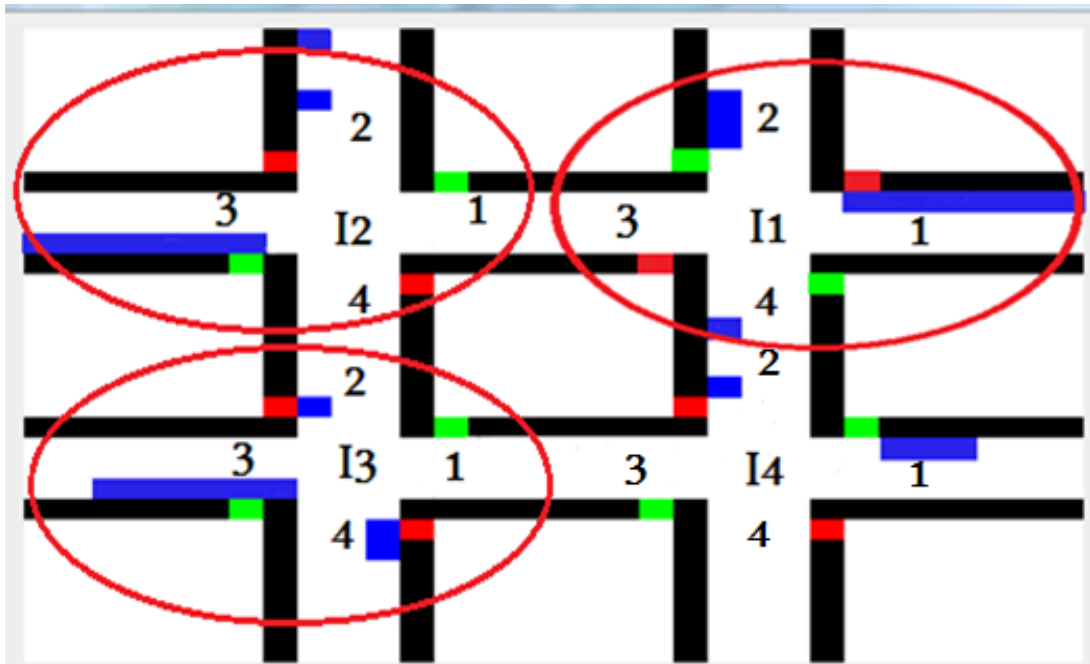


FIGURE 4.7 Cas de congestion , exemple 1

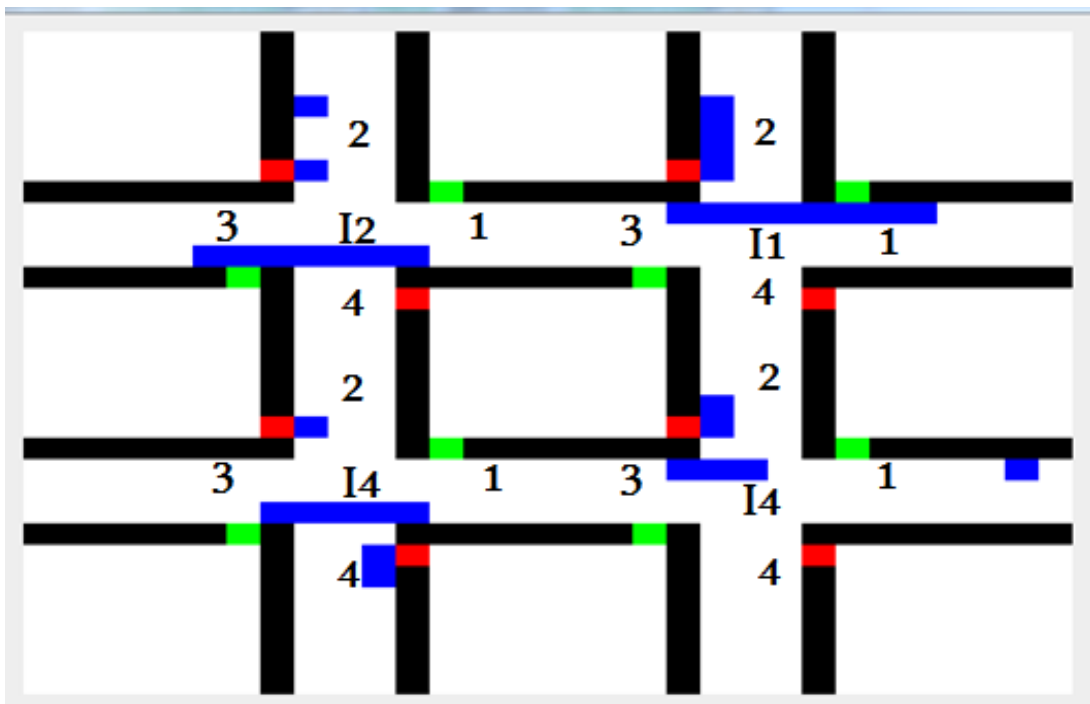


FIGURE 4.8 L'état e la route suite à la réponse du système.

Comme le montre la figure 4.7, on constate un embouteillage dans l'intersection I1 sur la route 1. Après application de la solution calculée par notre système, la route commence à se dégager en faisant passer la plupart des véhicules qui étaient en attente devant le feu rouge figure 4.8.

6.1.1.Exemple 2

La figure 4.9 représente un deuxième état de la route soulevant des problèmes de congestion sur les intersection I1, I2 et I3. Le nouvel état de la route dégagé suite à la réponse du système est celui donné par la figure 4.10. Il est clair que la bonne décision a été prise par le système en affectant le temps vert permettant de dégorger la situation et de résoudre même partiellement le problème de congestion

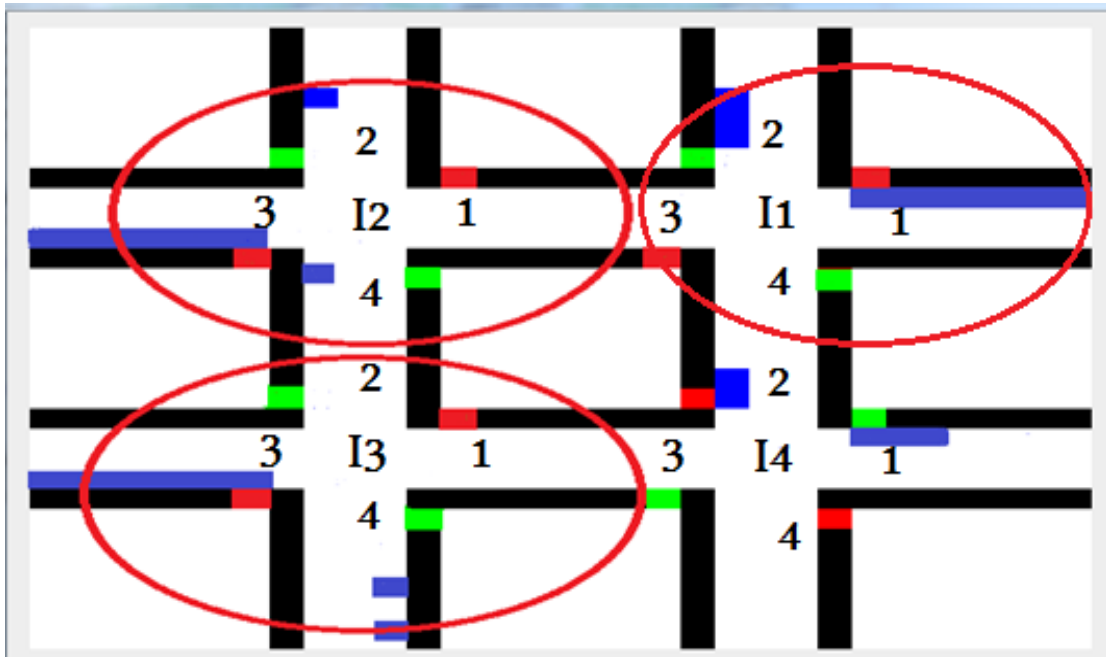


FIGURE 4.9 Cas de congestion, exemple 2

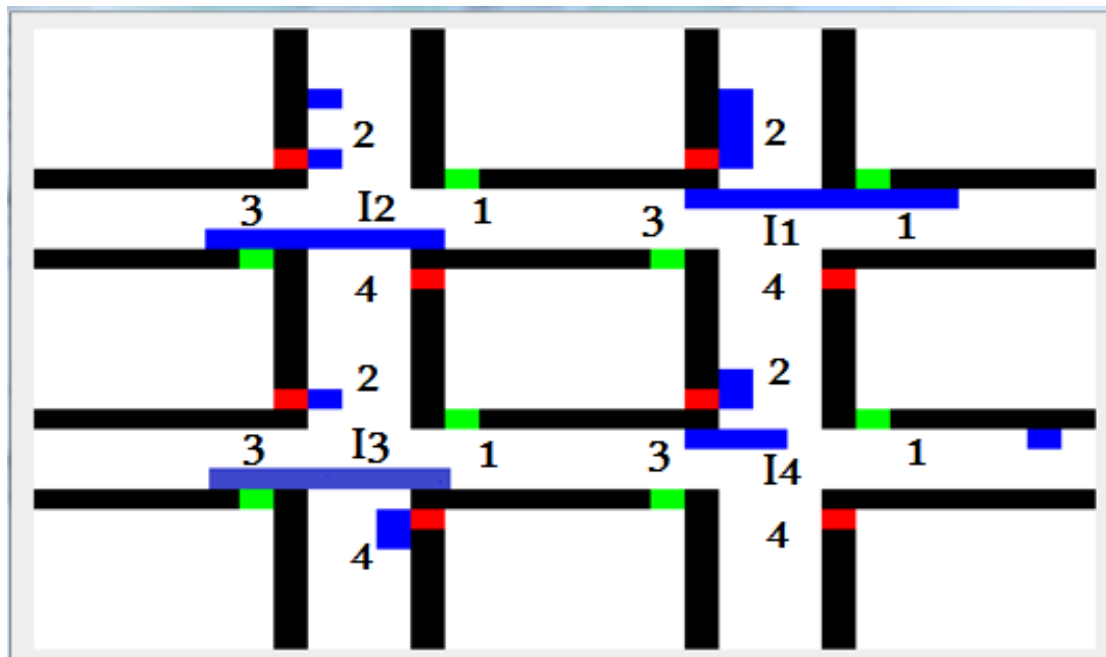


FIGURE 4.10 Etat de la route suite à la réponse du système.

## 6.2. Discussion

Le système proposé a bénéficié des avantages de la technique d'optimisation exploitée qui est celle des algorithmes génétiques en étant d'utilisation simple où il suffit d'introduire, une seule fois lors de la première exécution du système, la combinaison initiale des temps verts pour avoir la proposition optimale calculée par le système. De même le résultat sera obtenu en un temps de calcul très raisonnable par rapport aux méthodes exactes. Toutefois, on a pas pu transposer des exemples de temps de calcul du système proposé car il s'agit d'une simulation.

Parlons toujours des caractéristiques du processus génétique, et en considérant cette fois-ci le codage, celui proposé (codage proposé) dote le système d'une dynamique et adaptabilité de la combinaison des temps verts associés aux différentes routes des quatre intersections. En effet, grâce au codage binaire utilisé de nouvelles valeurs de temps de feux verts peuvent apparaître à chaque exécution en permettant, ainsi, de s'adapter aux nouvelles situations et exigences de la route. Par contre, Le système proposé par [45], exploite le même codage mais opère sur des valeurs entières de temps. De ce fait, le système ne fait que "tourner" les mêmes valeurs de temps sur les différentes routes d'intersections en générant à chaque fois de nouvelles combinaisons formées des mêmes valeurs de temps. Chose qui n'ai pas du tout logique.

En outre, le système que nous avons présenté peut être utilisé dans le cadre des plateformes de villes intelligentes pour gérer le système de transport grâce au contrôle des feux de circulation intelligents. Le système dépend de certaines théories et règles qui ont fait ordonner la priorité et l'intervalle de lumière verte.

## 7. Conclusion

Dans ce chapitre, nous avons présenté la structure de l'application pour simuler notre système proposé que nous avons développé à l'aide du langage de programmation JAVA. Le paramétrage adopté a été aussi présenté suivi d'une illustration des principales étapes du processus génétique de développement à savoir la génération de la population initiale et l'application des opérateurs de reproduction (croisement / mutation). En dernier lieu, des exemples des résultats obtenus par la simulation du système proposé ont été discutés.

## CONCLUSION GENERALE

**L**es méthodes modernes de gestion du trafic aux intersections (principalement, les feux de circulation faisant l'objet de notre étude) améliorent dans une large mesure les conditions de circulation.

Le but de notre travail présenté à travers ce mémoire était de concevoir et de développer un nouveau système intelligent de gestion des feux de circulation exploitant l'heuristique des algorithmes génétiques choisie, principalement, à cause de sa large utilisation du caractère aléatoire très prometteur et très intéressant à exploiter dans un tel domaine et donnant lieu, ainsi, à un système dynamique adaptables aux changements des états de circulation.

Ainsi, en ayant connaissance du nombre de véhicules et du temps moyen d'attente à une intersection, il nous sera possible d'estimer des temps de feux verts formant la première composante correspondant au chromosome initial à partir duquel notre système calcule, au fil des itérations, la meilleure combinaison de temps verts optimisant la gestion du trafic routier aux intersections.

Le système proposé a été testé sur des cas de congestion où la solution proposée a permis répondre à l'objectif de base qui est celui de résoudre le problème d'embouteillage aux intersections en étant dynamique et non pas fixe, c'est-à-dire que le temps vert accordé à une route varie suivant le nombre de véhicules en attente sur le feu.

Toutefois et afin de préparer les données en entrée du processus génétique, certaines caractéristiques du trafic doivent être déterminées en temps réel. En général, les caractéristiques nécessaires sont: le nombre de véhicules passant sur la route; Vitesse de ces véhicules. C'est pourquoi nous suggérons d'ajouter un module d'alimentation en données en temps réel à intervalles réguliers soit par utilisation : de capteurs magnétiques [46], de capteurs infrarouges [47] et de caméras [48] pour recueillir des caractéristiques de circulation.

## *Bibliographie*

- [1] R. JACOME , G.Riofrio , M. Augusto Article . Une enquête sur les feux de signalisation intelligents. Article uploaded by Manuel Augusto Pesantez Gonzalez , Octobre 2018.
- [2] I.ALIOUA ,et A.DJOUDER . Gestion de Trafic Urbain à base de Réseau de Capteurs sans Fil : Cas de la Ville de Bejaia . Mémoire de fin de cycle en vue de l'obtention du diplôme de Master en Recherche Opérationnelle, 2012
- [3] OCDE ,Gérer la congestion urbaine, Éditions OCDE, Disponible sur <http://dx.doi.org/10.1787/9789282101315-fr> , (2010).
- [4] THAI-PHU NGUYEN , Conception et application d'un modèle de l'information routière et ses effets sur le trafic .Thèse de doctorat, universite paris-est .Page 19, Disponible sur : <https://pastel.archives-ouvertes.fr/tel-00626631/document> , 2010.
- [5] Arun Kumar , Développement d'un système intelligent de feux de circulation à l'aide Réseau à tolérance de retard. Thèse en exécution partielle pour l'attribution du diplôme de maître de la technologie dans informatique ,université guru gobind singh indraprastha 2010-2012.
- [6] Marianne Vanderschuren , Andrew Mckune , Intelligent Transport Systems thèses Disponible sur : [https://www.researchgate.net/publication/283892659\\_Intelligent\\_Transport\\_Systems](https://www.researchgate.net/publication/283892659_Intelligent_Transport_Systems) ,2011.
- [7] S. FAYE. Contrôle du trafic routier urbain par un réseau fixe de capteurs sans fil. Thèse P.H.D, LTCl. Décembre 2011.
- [8] Benyamina Zakaria, Bounaama Fateh , Benahmed khelifa , Les systemes de transport intelligent(STI),disponiblesur:[https://www.researchgate.net/publication/333811970\\_Les\\_systemes\\_de\\_transport\\_intelligent\\_STI](https://www.researchgate.net/publication/333811970_Les_systemes_de_transport_intelligent_STI), 2017.
- [9] B. ZHOU, J. CAO, X. ZENG, and H. WU. Adaptive traffic light control in wireless sensor network-based intelligent transportation system. In Vehicular Technology Conference Fall (VTC 2010-Fall), pages 1-5.
- [10] D. ROBERTSON and R. BRETHERTON. Optimizing networks of traffic signals in real time-the scoot method. IEEE Transactions on Vehicular Techno-logy, 40(1) :11 -15. Feb. 1991

- [11]** B. M. CHARD and C. J. LINES. Transyt - the latest developments. Traffic engineering and control, 28(7-8) :387-390, 1987.
- [12]** Association mondiale de la route / World Road Association. AIPCR Manuel sur les systèmes de transport intelligents (STI) (seconde édition).Anglaise : ISBN 2-84060-174-5 Route 2 Market Ltd. Française : ISBN 2-84060-188-5AIPCR Secrétariat. Paris, 2003.
- [13]** M. SELINGER and L. SCHMIDT. A review of the cost, maintenance and reliability of popular adaptive traffic control technologies. Adaptive Traffic Control Systems in the United States. Sep 2009.
- [14]** N. HOUNSELL, B. SHRESTHA, F. MCLEOD, S. PALMER, T. BOWEN, and J. HEAD. Using global positioning system for bus priority in london : traffic signals close to bus stops. Intelligent Transport Systems, IET, 1(2) :131-137. June 2007.
- [15]** K. L. HEAD, P. B. MIRCHANDANI, and D. SHEPPARD. Hierarchical framework for real-time traffic control. Transportation Research Record 1360,Transportation Research Board, National Research Council, Washington DC 1992
- [16]** F. BOILLOT, J. BLOSSEVILLE, J. LESORT, V. MOTYKA, M. PAPA-GEORGIU, and S. SELLA. Optimal signal control of urban traffic networks. In 6th International Conference on Road Traffic Monitoring and Control, number 472, pages 182-186., 1992.
- [17]** J. HENRY, J. FARGES, and J. TUFFAL. The prodyn real time traffic algorithm. In 4th Conference on Control Transportation System, numer 472, pages 305-309., 1983C. REGGIE and G.CHRIS. Insync adaptive traffic signal technology : Real-time artificial intelligence delivering real-world results. July 2010.
- [18]** M. DUCARNE and T. PERINEAU. Commande adaptative et optimale d'un carrefour à feu. Technical report., 2007.
- [19]** C. LEE. Fuzzy logic in control systems : fuzzy logic controller. Part ii. IEEE Transactions on Systems, Man and Cybernetics, 20(2) :419-435. 1990.
- [20]** Z. LIAO and L. ZHAO. Wireless sensor networks help to improve the traffic safety in residential communities. In 6th International Conference on ITS Telecommunications Proceedings, pages 973 -978. June 2006.
- [21]** C. M. LESCIEUX. Application à la commande floue. <http://auto.polytech.univtours.fr/automatique/AUA/ressources/Introductionlogiquefloue.ppt>.2006.

- [22] K. YOUSEF, J. AL-KARAKI, and A. SHATNAWI. Intelligent traffic light flow control system using wireless sensors networks. *Journal of Information Science and Engineering*, 26(3) :753-768., 2010.
- [23] X.F. CHEN et Z.-K. SHI. Real-coded genetic algorithm for signal timing optimization of a single intersection. In *International Conference on Machine Learning and Cybernetics*, volume 3, pages 1245 - 1248, 2002.
- [24] S. TAKAHASHI, H. NAKAMURA, H. KAZAMA, and T. FUJIKURA. Genetic algorithm approach for adaptive offset optimization for the fluctuation of traffic flow. In *The IEEE 5th International Conference on Intelligent Transportation Systems*, pages 768 - 772., 2002.
- [25] I.C. LERMAN, R.F. NGOUENET. Algorithmes génétiques séquentiels pour une représentation affine des proximités, Rapport de recherche N° 2570, Institut National de recherche en informatique et en automatique (INRIA) ,66p, Disponible sur : <https://hal.inria.fr/inria-00074111>, 2006.
- [26] A.SOUQUET and F.G.RADET . Algorithmes génétiques, TE de fin d'année, 50 p. Université de Nice Sophia Antipolis, Disponible sur : [http://souqueta.free.fr/Project/files/TE\\_AG.pdf](http://souqueta.free.fr/Project/files/TE_AG.pdf) , 2004.
- [27] P.JAUFFRET. Une introduction aux algorithmes génétiques, 21p. Disponible sur : [https://masterchemoinfo.u-strasbg.fr/Documents/Fouille\\_de\\_donnees/Cours/algo\\_genetique.pdf](https://masterchemoinfo.u-strasbg.fr/Documents/Fouille_de_donnees/Cours/algo_genetique.pdf), 2002.
- [28] C. COELLO , A. CARLOS , D.A. VAN VELDHUIZEN , et GB . LAMONTG.B. EVOLUTIONARY algorithms for solving multi-objective problems, Kluwer academic publisher, New York, 576 p, 2002.
- [29] D.E.GOLDBERG. Algorithmes génétiques exploration, optimisation et apprentissage automatique , Éditions Addison -Wesley, France, 417 p,1994.
- [30] P. R .JEAN, . Introduction aux algorithmes génétiques. <http://www.renard.org/alife/french/gavintr.html> , 2000.
- [31] L. KARTOUBI. Optimisation de la synthèse des FACTS par les algorithmes génétiques et les essaims particulaires pour le contrôle des réseaux électriques. Mémoire de magister, Ecole Nationale Polytechnique, Alger, 86p, 2006.
- [32] A. AYADI. Application d'une variante d'algorithmes génétiques, Mémoire de magister, Université de Msila, 91 p , 2012.

- [33] Z.MICHALEWICZ , K.DEB, M. SHMIDT , et TH. STIDSEN .Evolutionary algorithms for engineering applications, pp.73-94. Dans “Evolutionary algorithms in engineering and computer science: recent advances in genetic algorithms, evolution strategies, evolutionary programming, genetic programming and industrial applications”, Edité par: Makela M., MiettinenK.,Neittaanmaki P. et Periaux J., Chichester, New York : Wiley, 500p, .1999.
- [34] M. TOMASSINI.(1999). Parallel and distributed algorithms : review, pp 113-133, dans: ”Evolutionary algorithms in engineering And computer science: recent advances in genetic algorithms,evolution strategies, evolutionary programming, genetic programming and industrial applications”, Edité par Markela M., Miettinen K.,Neittaanmaki P. et Periaux J., Chichester, New York : Weley, 500p, 1999.
- [35] D.E .GOLDBERG. Algorithmes génétiques exploration, optimisation et apprentissage automatique, Éditions Addison-Wesley, France, 417 p, 1994.
- [36] J. CRISPIN , H. HERNANDEZ . Algorithmes métaheuristiques hybrides pour la Sélection de gènes et la classification de données de biopuces, Thèse doctorat, Ecole doctorale Stim, 135p. Disponible sur : <https://tel.archives-ouvertes.fr/tel-00447684> , 2010.
- [37] A. SOUQUET , and F.G. RADET . Algorithmes génétiques, TE de fin d’année, 50 p. Université de Nice Sophia Antipolis, Disponible sur: [http://souqueta.free.fr/Project/files/TE\\_AG.pdf](http://souqueta.free.fr/Project/files/TE_AG.pdf) , 2004.
- [38] P .WILDI-TREMBLAY . Optimisation d’un matériau poreux stratifié pour un refroidissement maximal en convection forcée à l’aide d’un algorithme génétique, Mémoire de maîtrise, Université Laval Québec, 72p, 2006.
- [39] F.A AZOUD , M.W.BELHADJI , W. BENCHAIB , S. BOUHENNI. et S. SI-MOUSSI . Article . Algorithmes génétiques sophistiqué et schémas de parallélisation sur une grille de calcul pour la résolution du problème de sac à dos (Unbounded Knapsack),18 p. Disponible sur : <https://www.researchgate.net/publication/311442211> , Juin 2016.
- [40] L. KARTOUBI . Optimisation de la synthèse des FACTS par les algorithmes génétiques et les essaims particuliers pour le contrôle des réseaux électriques. Mémoire de magister, Ecole Nationale Polytechnique, Alger, 86p, 2006.
- [41] P. COLLET , P.R. JEAN . Handbook of Research on Nature Inspired Computing for Economics and Management :Stochastic Optimization Algorithms ,Edition Rennard ,J.P, Hershey, IGR , 2006.



- [42] M. HADJ-RACHID , C. BLOCH , C. RAMDANE . et P. CHATONNAY . Différentes opérateurs évolutionnaires de permutation : sélection, croisement et mutation, Rapport de recherche, Université de Franche Comté de Montbéliard, 60p, 2010.
- [43] C.YVES, J. OHLER ,S. TOLLARI. Algorithmes génétiques pour résoudre le problème du commis voyageur. Disponible sur: <http://sis.univtl.fr/~tollari/TER/AlgoGen1/node5.html#SECTION>, 2002.
- [44] M. RENVERSADE . Optimisation d'un dispositif hyper-sustentateur par Renversade, M. Optimisation d'un dispositif hyper-sustentateur par <https://hal.inria.fr/inria-00072610> , 2006.
- [45] S.M. Odeh . Article. Management of An Intelligent traffic Light System by using genetic algorithm , june 2013.
- [46] Q. WANG, J. ZHENG, H. XU, B. XU, et R. CHEN, Road-side magnetic sensor system for vehicle detection in urban environments, IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 5, pp. 1365–1374,May 2018.
- [47] L. P. J. RANI, M. K. KUMAR, K. S. NARESH, et S. VIGNESH, Dynamic traffic management system using infrared (IR)and Internet of Things (IoT), 3rd IEEE International Conference on Science Technology, Engineering and Management, pp. 353–357, 2018.
- [48] W.C. TCHUITCHEU,C. BOBDA et M.J.H. PANTHO. Internet of smart-cameras for traffic lights optimization in smart cities, Internet of Things 11 (2020) 100207, 2020.