

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne Démocratique et Populaire  
وزارة التعليم العالي والبحث العلمي  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



N° Réf :.....

Centre Universitaire  
Abd Elhafid Boussouf Mila

Institut des Sciences et Technologie

Département de Mathématiques et Informatique

## Mémoire préparé en vue de l'obtention du diplôme de Master

En : Informatique  
Spécialité: Sciences et Technologies de l'Information et de la  
Communication (STIC)

### Thème :

Optimisation du trafic urbain via contrôle des feux de  
signalisation, basée stratégie d'apprentissage interactif  
entre essaim de particules

Préparé par : Adjeroud Nada  
Belkhir Hala

Soutenu devant le jury

Encadré par Talai Meriem  
Président Deffas Zeyneb  
Examineur Selman Samir

Année Universitaire : 2020/2021

# Remerciement

---

*Tout d'abord, louanges et remerciements à Dieu tout-puissant, qui nous a permis d'achever ce travail ; Dieu merci.*

*Toutes les expressions de remerciements et d'appréciation à notre professeure et encadreur Madame « Talai Meriem », pour les efforts qu'elle a faits avec nous en termes d'orientations, de conseils, et pour sa patience avec nous ; Merci beaucoup.*

*Tous les remerciements à ma chère amie et voisine « Roumaïssa », qui nous aidée beaucoup.*

*Merci à tous nos professeurs dans notre cheminement académique.*

*On n'oublie pas de remercier nos parents, nos frères et sœurs, nos amis et collègues pour leur soutien moral et physique.*

# Dédicaces

---

*Nada :*

*Je dédie ce modeste travail à :*

*Premièrement et avant tous, à mes chères parents pour leurs soutient, leurs patience et leurs amour qui m'ont données la force pour continuer mes études : l'homme de ma vie et mon source d'espoir. Le meilleur homme et père dans ce monde « Adjeroud Mohammed\_Tayeb » qui je lui dire : « merci beaucoup papa ».*

*À la femme de ma vie, « Belkhane Hadjira » qui je lui dire : « Merci beaucoup Mama ».*

*À mes frères, de grand à le petit : « Soufian » et sa femme « Meriem », « Mourad », « Hicham » et « Kamel » ; qui sont la source de mes rires et mes inspirations.*

*À la plus jolie file dans ce monde, ma petite papillon « Amina Maram », et son petit beau-frère « seif-eddin ».*

*À tous mes cousines et mes soeurs : « Anfal », « Lamyia », « Najwa », « Amani », « Sabrina ».*

*À mes meilleures et chères amis : « Ikram », « Fatiha », « Lamyia », « Maissa », « Roumaissa », « Khouloud », « Mouna », « Hasna » et « Meriem ».*

*À Ma collègue « Hala ».*

*À Mes collègues de Classe de « Master Stic 2020 » : « Nassiba », « Malak », « Chaima », « Amina », « Hasna »,..., nous avons réuni les meilleurs moments et les plus beaux souvenirs.*

*À des collègues de classe « Master Stic 2021 ».*

*Merci à tous ceux qui m'ont soutenu même avec un mot gentil, un petit sourire qui a insufflé de l'espoir dans mon cœur.*

*Hala :*

*Je dédie ce modeste travail à :*

*Il est naturel que ma pensée la plus forte aille vers ma mère « Belkhir Ghania », à qui je dois la vie et une part essentielle de ma personnalité. Qu'elle sache que l'amour qu'elle me donne continue à m'animer et me permet d'envisager l'avenir comme un défi.*

*Ce travail est dédié à mon père « Belkhir Youcef », décédé trop tôt, qui m'a toujours poussé et motivé dans mes études. J'espère que, du monde qui est sien maintenant, il apprécie cet humble geste comme preuve de reconnaissance de la part d'un fils qui a toujours prié pour le salut de son âme. Puisse Dieu, le tout puissant, l'avoir en sa sainte miséricorde*

*À mes frères, et mes sœurs, de grand à le petit : « Badis » et « Amine », et « Rahim » et « Sohila », « Mona », « Lwiza » ; qui sont la source de mes rires et mes inspirations.*

*À tous les petits beaux enfants de mes frères et mes sœurs.*

*À tous mes cousines*

*À tous mes amies*

*À Ma collègue « Nada ».*

*À des collègues de classe « Master Stic 2021 ».*

*Merci à tous ceux qui m'ont soutenu même avec un mot gentil, un petit sourire qui a insufflé de l'espoir dans mon cœur.*

# Résumé

---

Ce mémoire traite l'optimisation du trafic urbain via contrôle des feux de signalisation, basée sur la stratégie d'apprentissage interactif entre essaim de particules. Cela comprend l'optimisation de la durée des phases sur quelques feux dans la zone urbaine de la wilaya de Mila en tenant compte du flux réel de véhicules. La simulation du trafic est un processus indispensable qui permet de définir une série d'arrivées des véhicules sur le réseau, de déployer la variante du PSO pour contrôler le réseau de circulation des feux de signalisation et de récupérer les traces pour mesurer l'efficacité de la solution au trafic routier. Les résultats ont montré sa capacité à réduire les temps d'attente des véhicules et ainsi réduire les embouteillages et la pollution de l'air dans la wilaya de Mila.

# Abstract

---

This thesis deals with the optimization of urban traffic through traffic light control, based on the interactive learning strategy between swarms particle. This includes the optimization of the duration of the phases on some fires in the urban area of the wilaya of Mila, taking into account the real flow of vehicles. Traffic simulation is an indispensable process which allows to define a series of vehicle arrivals on the network, to deploy the variant of the PSO to control the traffic network of the traffic lights and to retrieve the tracks to measure the efficiency of the traffic. the solution to road traffic. The results showed its ability to reduce vehicle waiting times and thus reduce traffic jams and air pollution in the wilaya of Mila.

## ملخص

---

تعالج هذه المذكرة تحسين حركة المرور في المناطق الحضرية من خلال التحكم في إشارات المرور بناءً على استراتيجيات التعلم التفاعلي بين أسراب الجسيمات. ويشمل ذلك تحسين مدة المراحل لبعض إشارات المرور في المنطقة الحضرية لولاية ميلت، مع مراعاة التدفق الحقيقي للمركبات. محاكاة حركة المرور هي عملية لا غنى عنها والتي تسمح بتحديد سلسلة من وصول المركبات على الشبكة، لتوظيف متغير تحسين سرب الجسيمات للتحكم في شبكة إشارات المرور واستعادة المسارات لقياس كفاءة حركة المرور. وأظهرت النتائج قدرتها على تقليل أوقات انتظار المركبات وبالتالي تقليل الاختناقات المرورية وتلوث الهواء في ولاية ميلت.

# Table de matière

---

Introduction général.....	A
---------------------------	---

## **Chapitre 1 : Optimisation par essaim particulaire**

1. Introduction.....	1
2. Classes des problèmes d’optimisations .....	1
2.1 Optimisation mono-objectif.....	1
2.2 Optimisation Multi-objectif.....	2
3. Problèmes d’optimisation multi-objectif.....	2
3.1 Formalisation .....	2
3.2 Principale caractéristique.....	3
3.2.1 Approche Pareto.....	3
3.2.2 Notion de dominance .....	3
3.3 Approches de résolution .....	4
3.3.1 Classification « point de vue décideur » .....	4
3.3.2 Classification « point de vue concepteur » .....	5
4. Les méthodes d’optimisation .....	5
4.1 Principales catégories des méthodes d’optimisation.....	5
4.2 Les heuristiques et les Méta-heuristiques :.....	6
4.3 Extensions et application des méta-heuristiques .....	7
5. Méthode d’optimisation par essaim particulaire .....	8
5.1 Principes .....	8
5.2 Formalisation .....	9
5.3 Amélioration de PSO.....	11
5.3.1 Confinement des particules .....	11
5.3.2 Coefficient de constriction .....	12
5.3.3 Coefficient d’inertie .....	13
5.3.4 Topologie de voisinage : .....	13
6. Optimisation par trois essaims particulaires coopératives .....	14
6.1 Principe générale .....	14
6.2 L’approche coopérative des trois essaims.....	16
7. Conclusion.....	17

## **Chapitre 2 : Généralités sur la modélisation et le contrôle du trafic routier**

1. Introduction.....	18
2. Les variables de trafic .....	19
2.1 Véhicules considéré individuellement.....	19
2.1.1 Notion de trajectoire.....	19
2.1.2 Mesures de la position longitudinale.....	19
2.1.3 Mesures de $t(x)$ .....	20
2.1.4 Variables et caractéristiques d'un véhicule.....	21
2.1.5 Temps de parcours individuel .....	21
2.2 Variables concernant deux véhicules successifs.....	22
2.3 Variables Concernant un flot de véhicules .....	23
3. Modélisations du trafic routier .....	25
3.1 Modèles microscopiques : .....	26
3.2 Modèles macroscopiques .....	27
3.3 Modèles mésoscopiques .....	27
4. Contrôle de feux de signalisation .....	27
4.1 Objectifs et domaine d'emploi .....	27
4.2 Principe .....	28
4.3 Stratégies de régulation.....	28
4.3.1 Stratégie multiprogrammation (prédéterminée ou cyclique).....	29
4.3.2 Stratégie micro-régulation(semi-adaptative ou acyclique).....	29
4.3.3 Stratégie temps réel (adaptative).....	29
4.4 Les critères à optimiser .....	30
5. Conclusion.....	31

## **Chapitre 3 : L'outil de simulation du trafic urbain SUMO**

1. Introduction.....	32
2. Simulation.....	32
3. Le simulateur SUMO.....	32
3.1 Principales fonctionnalités .....	33
3.2 Outils de SUMO .....	35
3.3 Lancement d'une simulation SUMO .....	36
3.4 Construction de Réseau .....	37
3.4.1 Création à partir des éléments de l'infrastructure .....	37
3.4.2 Importer une topologie à partir d'autres formats .....	40

3.5	Le model de traffic.....	40
3.5.1	Définir le trafic « à la main » .....	41
3.5.2	Spécification du trafic via outils de SUMO .....	42
3.6	Les résultats des simulations.....	43
3.7	Contrôle des simulations via TRACI.....	44
3.7.1	Utiliser TraCI : .....	44
3.7.2	Plusieurs clients : .....	44
6.	Conclusion.....	45

## **Chapitre 4 : Contrôle des feux tricolores via PSO à trois essais coopératifs sur SUMO**

1.	Vue générale du problème traité .....	46
2.	Formalisation du problème .....	47
3.	Stratégie d'optimisation .....	48
3.1	Résolution via PSO à 3 essais cooperatifs .....	49
3.1.1	Avantage du PSO .....	49
3.1.2	Description des particules .....	50
3.1.3	Paramètres du PSO.....	51
3.1.4	Etapas du TSC-PSO pour le contrôle proposé .....	52
4.	Mise en œuvre de SUMO dans la démarche de contrôle.....	54
4.1	Préparation des scénarios.....	54
4.2	Interaction entre SUMO et TSC-PSO.....	56
5.	Organigramme de code .....	58
7.	Statistiques sur les routes de Mila.....	60

## **Chapitre 5 : Tests et analyse des résultats**

1.	Introduction .....	63
2.	Démarche de test .....	63
3.	Analyse comparative .....	64
4.	Conclusion.....	67
	Conclusion général.....	68

# Table des figures

---

## Chapitre 1

Figure 1.1:Stratégie de déplacement d'une particule. [3] .....	9
Figure 1.2:Organigramme de la méthode PSO. [2].....	11
Figure 1.3:Topologies de voisinage.[4].....	14

## Chapitre 2

Figure 2.1:position sur la voie d'un véhicule à quatre instants différents. [7].....	19
Figure 2.2:trajectoire d'un véhicule dans le plan (x, t). [7].....	20
Figure 2.3: deux trajectoires de véhicules : à gauche (a/) une trajectoire interpolée entre deux mesures de position successives ; à droite (b/) à partir d'une mesure continue de la vitesse et de la position. [7].....	20
Figure 2.4: évolution d'un véhicule dont la position est mesurée en des points régulièrement espacés de la voie. [7] .....	21
Figure 2.5: trajectoires de deux véhicules : un véhicule leader et son véhicule suiveur. [7] ...	23
Figure 2.6: évolution de la proportion du débit sur chaque voie en fonction du débit total (en véh/h) supporté par une autoroute à trois voies (autoroute A7) . [7] .....	24
Figure 2.7: Représentation du débit Q et concentration K par rapport aux trajets de véhicules dans une route. [8].....	25
Figure 2.8: Diagrammes débit-densité .[7].....	26
Figure 2.9: classification des modèles de trafic.....	26

## Chapitre 3

Figure 3.1:SUMO.....	33
Figure 3.2: Vue de la ville de Mila en OpenStreeMap.....	34
Figure 3.3: Réseau importé dans SUMO.....	34
Figure 3.4: Exemple de « Sumocfg file » .....	37
Figure3. 5: Exemple des « nodes » .....	38
Figure 3.6: Exemple de « edges ». .....	38
Figure 3.7: Exemple de « jonction ». .....	38
Figure 3.8: Exemple de programme « tls » . .....	39
Figure 3.9: Exemple de programme de connexion.....	39
Figure 3.10: Exemple de NET file. ....	42
Figure 3.11: Exemple de Route file.....	43
Figure 3.12: Fichiers de configuration de la circulation urbaine.....	46

## Chapitre 4

Figure 4.1: Stratégie d'optimisation.....	49
Figure 4.2:Illustration d'une particule selon le problème étudié.....	51
Figure 4.3:Stratégie d'optimisation avec 3 essais particulaire.....	53
Figure 4: L'étape de permutation entre un mauvais et un excellent essaim.....	54

Figure 4.5: Intersection avec plusieurs feux de circulation sélectionnés à partir de la carte d'instance SUMO. ....	56
Figure 4.6: Communication entre SUMO et TSC-PSO via Traci .....	57
Figure 4.7: Organigramme principal. ....	58
Figure 4.8: Organigramme de fonction fitness_local. ....	59
Figure 4.9: Organigramme de fonction fitness. ....	60
Figure 4.10: Carte du réseau de Mila .....	61
Figure 4.11: Histogramme de déplacement. ....	62

## **Chapitre 5**

Figure 5.1: Exemple de fichier « summary ».....	64
Figure 5.2: Exemple d'un fichier trace « tripinfo ». ....	64
Figure 5.3: Comparaison du nombre de véhicules en déplacement sur le réseau. ....	65
Figure 5.4: Comparaison de temps moyen de déplacement des véhicules.....	66
Figure 5.5: Comparaison des temps d'attente moyens de véhicule.....	67

# Liste des tableaux

---

Tableau 5.1: Résultats d'exécution de TSC_PSO avec plusieurs valeurs différentes des coefficients d'accélération.....	63
-----------------------------------------------------------------------------------------------------------------------	----

# Introduction Générale

---

L'urbanisation accélérée des populations ainsi que le développement des échanges socio-économiques dans les agglomérations ont eu pour conséquence une croissance rapide de la mobilité urbaine asphyxiant même plusieurs zones urbaines. En effet, les conséquences de cette croissance sont néfastes notamment l'allongement des congestions. Les ralentissements aux heures de pointe ont un impact notable sur l'organisation des activités et sur l'économie. Elles peuvent être désastreuses pour l'environnement dans la mesure où les congestions résultent en l'émission de quantités de plus en plus importantes des gaz à effet de serre.

La communauté des chercheurs scientifiques, notamment les ingénieurs en technologies de l'information et de la communication ont permis, en l'espace de quelques années, de mettre en œuvre des systèmes de transport intelligents. Ils ont la capacité extraordinaire de réagir aux variations du trafic en régulant dynamiquement les flux de véhicules. Les équipements tels que les feux de circulation deviennent alors des éléments actifs dans la réduction des embouteillages et dans la gestion de la sécurité, améliorant globalement l'expérience des usagers.

L'Algérie comme la plupart des pays du tiers monde connaît également ces problèmes du trafic routiers urbain et souffre de ses conséquences économiques à l'échelle du pays, des collectivités locales et des individus. De plus, contrairement aux pays développés, les stratégies pour gérer cette mobilité urbaine sont basées sur des formalismes et des méthodes très anciennes voir obsolètes. Alors que les pays développés investissent en termes de construction mais surtout par la mise en place de nouvelles stratégies pour réduire les congestions et réguler le trafic général. Les nouvelles technologies de l'information et de la communication ont proposé des études et solutions qui varient des systèmes multi-agents aux réseaux de petri et des méthodes meta-heuristique, ayant chacune des contextes et objectifs spécifiques.

Dans le cadre de notre projet de fin d'études, nous proposons une solution aux problèmes d'embouteillage en ville. Nous travaillons sur le contrôle en temps différé du trafic routier sur la wilaya de Mila. Nous agissons sur les feux de signalisation afin d'optimiser les temps d'attente des véhicules durant leurs circulations. La méthode d'optimisation utilisée est une variante du PSO qui utilise trois essais de natures différentes afin de maximiser les performances de recherche de solution optimale. L'objectif de cette variante étant de bien

survoler les zones du domaine des solutions candidates à la résolution optimale du problème. Chacune des solutions faisables est l'ensemble des phases des programmes suivis sur les feux de signalisation choisis. En effet, la solution opère sur quelques feux mais son efficacité est mesurée sur une zone géographique plus large.

La mise en œuvre de la méthode proposée pour contrôler le trafic routier étant inapplicable directement à la réalité, il est impératif de recourir aux outils de simulation du trafic routier. Nous avons implémenté la proposition sur l'outil SUMO connu dans les recherches scientifiques par la précision et la multitude de possibilités qu'il offre en termes de création des scénarios et la récupération des traces et résultats des simulations. Nous utilisons des statistiques provenant de la direction des transports pour mieux approcher nos simulations au cas réel.

Le présent mémoire est organisé en cinq chapitres, outre l'introduction, répartis comme suit :

Le premier chapitre : Optimisation par essaim particulière.

Le deuxième chapitre : Généralités sur la modélisation et le contrôle du trafic routier.

Le troisième chapitre : L'outil de simulation du trafic urbain SUMO.

Le quatrième chapitre : Contrôle des feux tricolores via PSO à trois essais coopératifs sur SUMO.

Le cinquième chapitre : Tests et analyse des résultats.

On termine le manuscrit par une conclusion générale et des perspectives à notre travail.

# Chapitre 1 : Optimisation par essaim particulière

---

## 1. Introduction

Les problèmes d'optimisation occupent une place importante dans la communauté des ingénieurs, des scientifiques et des décideurs. En effet, ce genre de problèmes intervient dans leurs domaines d'activité qui sont très divers, comme la conception de systèmes mécaniques, le traitement d'images, l'électronique ou la recherche opérationnelle. Résoudre un problème d'optimisation consiste à trouver la ou les meilleures solutions en minimisant et/ou maximisant la/les fonctions objectifs du problème posé, tout en satisfaisant un ensemble de conditions définies par l'utilisateur. L'espace de recherche est l'ensemble des solutions au problème, à la fois possibles et acceptables.

## 2. Classes des problèmes d'optimisations

En s'intéressant au nombre d'objectifs à optimiser, nous identifions deux classes de problèmes d'optimisation : mono-objectif et multi-objectifs

### 2.1 Optimisation mono-objectif

On dit qu'un problème est mono-objectif, lorsqu'un seul critère (objectif) est donné. Dans ce cas la solution optimale est clairement définie, c'est celle qui a le coût optimal (minimal ou maximal). De manière formelle, à chaque instance d'un tel problème est associé un ensemble  $W$  des solutions potentielles respectant certaines contraintes et une fonction d'objectif  $f$  qui associe à chaque solution admissible  $s$  de  $W$  une valeur  $f(s)$ . Résoudre l'instance du problème d'optimisation consiste à trouver la solution optimale  $s^*$  de  $W$  qui optimise (minimise ou maximise) la valeur de la fonction objectif. Pour le cas de la minimisation : le but est de trouver  $s^*$  de  $W$  tel que  $f(s^*) < f(s)$  pour tout élément  $s$  de  $W$ . Un problème de maximisation peut être défini de manière similaire. [1]

Dans la plupart des problèmes d'optimisation, des restrictions sont imposées par les caractéristiques du problème. Ces restrictions doivent être satisfaites afin de considérer une solution acceptable. Cet ensemble de restrictions, appelées contraintes, décrit les dépendances entre les variables de décision et les paramètres du problème. On formule usuellement ces contraintes  $c_j$  par un ensemble d'inégalités, ou d'égalités de la forme :  $c_j(x_1, x_2, \dots, x_n) \geq 0$ . [1]

## 2.2 Optimisation Multi-objectif

La plupart des problèmes d'optimisation réels sont décrits à l'aide de plusieurs objectifs ou critères souvent contradictoires devant être optimisés simultanément. [1]

Pour les problèmes n'incluant qu'un seul objectif, l'optimum recherché est clairement défini. Cependant, pour les problèmes d'optimisation multi-objectif, celui-ci doit être formalisé. En considérant rien que deux objectifs à minimiser, toute amélioration de l'un des objectifs se fait au détriment de l'autre et la solution optimale ou proche de l'optimum est un compromis entre les deux. Par exemple, dans l'achat d'une voiture d'occasion, la voiture idéale est celle qui est peu chère (critère économique) avec peu de kilomètres (critère qualitatif), il n'est pas évident de pouvoir regrouper en un seul objectif ces deux critères non commensurables. Nous allons donc devoir identifier les meilleurs compromis possibles suivant notre budget. En effet, pour un problème à deux objectifs contradictoires, la solution optimale cherchée est un ensemble de points correspondant aux meilleurs compromis possibles pour résoudre notre problème.

## 3. Problèmes d'optimisation multi-objectif

### 3.1 Formalisation

Un problème d'optimisation multi-objectif MOP (multi-objective optimization problem) peut être défini comme suit :

$$\text{Min } f(x) = f_1(x), f_2(x), \dots, f_n(x).$$

$$(MOP), \text{ Tel que: } x \in X.$$

Où  $n$  est le nombre d'objectifs ( $n \geq 2$ ),

$X$  est l'ensemble des solutions réalisable dans l'espace décisionnel, et  $x = (x_1, x_2, \dots, x_k) \in X$  est un vecteur représentant les variables de décision, dans le cas combinatoire  $x$  est un ensemble discret, à chaque solution  $x \in X$  est associé un vecteur objectif  $z \in Z$  sur la base d'un vecteur de fonction  $f$  tel que :

$$f: X \rightarrow Z,$$

$$f(x) = (f_1(x), f_2(x), \dots, f_m(x)),$$

où  $z = (z_1, z_2, \dots, z_m)$  et  $Z = f(X)$  représente l'ensemble des points réalisable. [1]

### 3.2 Principale caractéristique

Les problèmes multi-objectifs ont la particularité d'être beaucoup plus difficiles à traiter que leur équivalent mono-objectif. La difficulté réside dans l'absence d'une relation d'ordre total entre les solutions. Une solution peut être meilleure qu'une autre sur certains objectifs et moins bonne sur les autres. Donc il n'existe généralement pas une solution unique qui procure simultanément la solution optimale pour l'ensemble des objectifs. Voilà pourquoi le concept de solution optimale devient moins pertinent en optimisation multi-objectif. Dans ce cas la solution optimale ou de bonne qualité est un ensemble de solutions compromis entre les différents objectifs à optimiser.

Il est vital pour identifier ces meilleurs compromis de définir une relation d'ordre entre ces éléments. La plus célèbre et la plus utilisée est la relation de dominance au sens Pareto. L'ensemble des meilleurs compromis est appelé le front Pareto, la surface de compromis ou l'ensemble des solutions efficaces. Cet ensemble de solutions constitue un équilibre, dans le sens qu'aucune amélioration ne peut être faite sur un objectif sans dégradation d'au moins un autre objectif. La solution Pareto consiste à obtenir le front de Pareto. [1]

#### 3.2.1 Approche Pareto

Au 19<sup>ème</sup> Siècle, Vilfredo Pareto, un mathématicien Italien, formule le concept suivant : dans un problème multi-objectif, il existe un équilibre tel que l'on ne peut pas améliorer un objectif sans détériorer au moins un des autres objectifs. Les approches Pareto utilisent directement la notion de dominance dans la sélection des solutions générées. Le principal avantage de ces approches, c'est l'optimisation simultanée d'objectifs contradictoires. [1]

#### 3.2.2 Notion de dominance

Lorsque l'on cherche à obtenir une solution optimale à un problème d'optimisation Multi-objectif donné, on s'attend souvent à trouver une solution et une seule. En effet, on rencontre rarement ce cas de figure. La plupart de temps, on trouve une multitude de solutions, du fait que certaines des objectifs sont contradictoires.

Le décideur évalue généralement une solution par rapport à chacun des critères, et se positionne donc naturellement dans l'espace objectif, par contre au cas mono-objectif ou il existe un ordre total parmi les solutions réalisables il n'existe généralement pas de solution qui serait à la fois optimale pour chaque objectif, étant donnée la nature conflictuelle de ces derniers, ainsi une relation de dominance d'ordre partiel est généralement définie. [1]

En simplifiant, la résolution de problèmes multi-objectifs relève de deux disciplines assez différentes. En effet, résoudre un problème multi-objectif peut être divisé en deux phases :

- **La recherche des solutions de meilleur compromis** : C'est la phase d'optimisation Multi-objectif.
- **Le choix de la solution à retenir** : C'est la tâche du décideur qui, parmi l'ensemble des solutions de compromis, doit extraire celle(s) qu'il utilisera. On parle alors ici de décision multi-objective et cela fait appel à la théorie de la décision.

### 3.3 Approches de résolution

On distingue deux classes différentes des approches de résolution de problèmes multi-objectifs. Le premier classement adopte un point de vue décideur où les méthodes sont classées en fonction de leur utilisation. Le deuxième classement adopte un point de vue concepteur, les méthodes y sont triées de leur façon de traiter les fonctions objectives. Ainsi avant de se lancer dans la résolution d'un problème multi-objectif, il faut se poser la question du type d'approche de résolution à utiliser. [1]

#### 3.3.1 Classification « point de vue décideur »

Cette classification propose trois schémas possibles. Soit le décideur intervient dès le début de la définition du problème, en exprimant ses préférences, afin de transformer un problème multi-objectif en un problème simple objectif. Soit le décideur effectue son choix dans l'ensemble des solutions proposées par le solveur multi-objectif, de deux manières différentes. Les trois schémas sont alors :

##### 3.3.1.1 Les approches à priori

Le décideur intervient en aval du processus d'optimisation, pour définir la fonction d'agrégation modélisant le compromis que l'on désire faire entre les différents objectifs. Dans ce cas le décideur est supposé connaître a priori le poids de chaque objectif afin de les mélanger dans une fonction unique. Cela revient à résoudre un problème mono-objectif. Cependant dans la plupart des cas, le décideur ne peut pas exprimer clairement sa fonction d'utilité, parce que les différents objectifs sont non commensurables (exprimés dans des unités différentes). [1]

##### 3.3.1.2 Les approches interactives

Combinent de manière cyclique et incrémentale les processus de décision et d'optimisation, le décideur intervient de manière à modifier certaines variables ou contraintes afin de diriger le processus d'optimisation. Le décideur modifie ainsi interactivement le compromis entre ses préférences et les résultats. Cette approche permet donc de bien prendre en compte les préférences du décideur, mais nécessite sa présence tout au long du processus de recherche. [1]

### 3.3.1.3 Les approches a posteriori

Cherche à fournir au décideur un ensemble de bonnes solutions bien réparties. Il peut ensuite, au regard de l'ensemble des solutions, sélectionner celle qui lui semble la plus appropriée. Ainsi, il n'est plus nécessaire de modéliser les préférences du décideur (ce qui peut s'avérer être très difficile), mais il faut en contrepartie fournir un ensemble de solutions bien réparties, ce qui peut également être difficile et requérir un temps de calcul important (mais ne nécessite pas la présence du décideur).

### 3.3.2 Classification « point de vue concepteur »

Ce classement adopte un point de vue plus théorique articulé autour des notions d'agrégation et d'optimum Pareto. Les approches sont conçues de deux manières : les approches non Pareto et les approches Pareto.

#### 3.3.2.1 Les approche non Pareto

Ne traitent pas le problème comme un véritable problème multi-objectif. Elles cherchent à ramener le problème initial à un ou plusieurs problèmes mono-objectifs. [1]

#### 3.3.2.2 Les approche Pareto

Ne transforment pas les objectifs du problème, ceux-ci sont traités sans aucune distinction pendant la résolution. [1]

## 4. Les méthodes d'optimisation

### 4.1 Principales catégories des méthodes d'optimisation

Les méthodes de résolutions des problèmes d'optimisation varient entre méthodes exactes et méthodes approximatives (approchées) [2]:

**Méthodes exactes** : Ces méthodes calculent des solutions optimales aux problèmes avec garantie de leur optimalité. [2]

**Méthodes approximatives** : Utilisées pour aborder des problèmes de grande taille et permettre d'obtenir des solutions de bonne qualité en des délais raisonnables. Ces méthodes sont subdivisées en deux sous classes [2] : Les heuristiques et les Méta-heuristiques.

### 4.2 Les heuristiques et les Méta-heuristiques :

Certains problèmes d'optimisation sont qualifiés de difficiles, et leur résolution dans un temps raisonnable nécessite l'utilisation d'algorithmes sophistiqués, comme les méthodes approchées (Les heuristiques et les méta-heuristiques). [3]

**Les heuristiques** sont des algorithmes approchés qui permettent d'identifier en temps polynomial au moins une solution réalisable rapide, pas obligatoirement optimale. Généralement sont conçue pour des problèmes particuliers, en s'appuyant sur sa structure propre sans offrir aucune garantie quant à la qualité de la solution calculée. [2]

**Les Méta-heuristiques** sont des heuristiques plus poussées, adaptables à un grand nombre de problèmes différents, sans changements majeurs dans l'algorithme, ont été mises au point et ont donné naissance à une nouvelle famille d'algorithmes d'optimisation stochastiques : les méta-heuristiques. Le terme méta-heuristique a été inventé par Fred Glover en 1986, lors de la conception de la recherche tabou.

Les méta-heuristiques forment une famille d'algorithmes d'optimisation visant à résoudre des problèmes d'optimisation difficile, pour lesquels nous ne connaissons pas de méthodes efficaces. Elles sont généralement utilisées comme des méthodes génériques pouvant optimiser une large gamme de problèmes différents, d'où le qualificatif méta. Leur capacité à optimiser un problème à partir d'un nombre minimal d'informations est contre balancée par le fait qu'elles n'offrent aucune garantie quant à l'optimalité de la meilleure solution trouvée. Cependant, du point de vue de la recherche opérationnelle, ce constat n'est pas forcément un désavantage, puisque l'on préfère toujours une approximation de l'optimum global trouvée rapidement à une valeur exacte trouvée dans un temps rédhibitoire. Il existe un grand nombre de méta-heuristiques différentes, allant de la simple recherche locale à des algorithmes complexes de recherche globale. La plupart des méta-heuristiques utilisent des processus aléatoires comme moyens de récolter de l'information et de faire face à des problèmes comme l'explosion combinatoire. [3]

Les méta-heuristiques peuvent être considérées comme des algorithmes stochastiques itératifs, où elles manipulent une ou plusieurs solutions à la recherche de l'optimum. Les itérations successives doivent permettre de passer d'une solution de mauvaise qualité à la solution optimale. L'algorithme s'arrête après avoir atteint un critère d'arrêt, consistant généralement en l'atteinte du temps d'exécution imparti ou en une précision demandée. Ces méthodes tirent leur intérêt de leur capacité à éviter les optima locaux, soit en acceptant des dégradations de la fonction objectif au cours du traitement, soit en utilisant une population de points comme méthode de recherche.

Les méta-heuristiques sont souvent inspirées de processus naturels qui relèvent de la physique (l'algorithme du recuit simulé), de la biologie de l'évolution (les algorithmes génétiques) ou encore de l'éthologie (les algorithmes de colonies de fourmis ou l'optimisation par essaim particulaire).

Parmi les méta-heuristiques citées, l'optimisation par essaim particulaire (OEP, ou PSO en anglais) est apparue en 1995. C'est la méthode utilisée pour notre travail.

### **4.3 Extensions et application des méta-heuristiques**

Les méta-heuristiques se caractérisant par leur capacité à résoudre des problèmes très divers, elles se prêtent naturellement à des extensions. L'auteur de (3) cite quelques une de ces extensions :

- Les méta-heuristiques pour l'optimisation multi-objectif : où il faut optimiser plusieurs objectifs contradictoires. Le but ne consiste pas ici à trouver un optimum global, mais à trouver un ensemble d'optima, qui forment une surface de compromis pour les différents objectifs du problème;
- Les méta-heuristiques pour l'optimisation multimodale : où l'on ne cherche plus l'optimum global, mais l'ensemble des meilleurs optima globaux et/ou locaux ;
- Les méta-heuristiques pour l'optimisation de problèmes bruités : où il existe une incertitude sur le calcul de la fonction objectif, dont il faut tenir compte dans la recherche de l'optimum ;
- Les méta-heuristiques pour l'optimisation dynamique : où la fonction objective varie dans le temps, ce qui nécessite d'approcher l'optimum à chaque pas de temps.
- Les méta-heuristiques hybrides : qui consistent à combiner différentes méta-heuristiques, afin de tirer profit des avantages respectifs.
- Les méta-heuristiques parallèles : où l'on cherche à accélérer le calcul, en répartissant la charge de calcul sur des unités fonctionnant de concert. Le problème revient alors à adapter les méta-heuristiques pour qu'elles soient distribuées. [3]

### 5. Méthode d'optimisation par essaim particulaire

#### 5.1 Principes

L'optimisation par essaim particulaire (OEP), ou Particule Swarm Optimisation (PSO) en anglais, est un algorithme qui utilise une population de solutions candidates pour développer une solution optimale au problème. Cet algorithme a été proposé par Russel Eberhart (ingénieur en électricité) et James Kennedy (socio-psychologue) en 1995 [Kenn 95].

Il s'inspire à l'origine du monde du vivant, plus précisément du comportement social des animaux évoluant en essaim, tels que les bancs de poissons et les vols groupés d'oiseaux. En effet, on peut observer chez ces animaux des dynamiques de déplacement relativement complexes, alors qu'individuellement chaque individu a une « intelligence » limitée, et ne dispose que d'une connaissance locale de sa situation dans l'essaim. L'information locale et la mémoire de chaque individu sont utilisées pour décider de son déplacement. Des règles simples, telles que « rester proche des autres individus », « aller dans une même direction » ou « aller à la même vitesse », suffisent pour maintenir la cohésion de l'essaim, et permettent la mise en œuvre de comportements collectifs complexes et adaptatifs.

L'essaim de particules correspond à une population d'agents simples, appelés particules. Chaque particule est considérée comme une solution du problème, où elle possède une position (le vecteur solution) et une vitesse. De plus, chaque particule possède une mémoire lui permettant de se souvenir de sa meilleure performance (en position et en valeur) et de la meilleure performance atteinte par les particules « voisines » (informatrices) : chaque particule dispose en effet d'un groupe d'informatrices, historiquement appelé son voisinage. Un essaim de particules, qui sont des solutions potentielles au problème d'optimisation, « survole » l'espace de recherche, à la recherche de l'optimum global. Le déplacement d'une particule est influencé par les trois composantes suivantes [3] :

**Une composante d'inertie** : la particule tend à suivre sa direction courante de déplacement ;

**Une composante cognitive** : la particule tend à se diriger vers le meilleur site par lequel elle est déjà passée ;

**Une composante sociale** : la particule tend à se fier à l'expérience de ses congénères et, ainsi, à se diriger vers le meilleur site déjà atteint par ses voisins. [3]

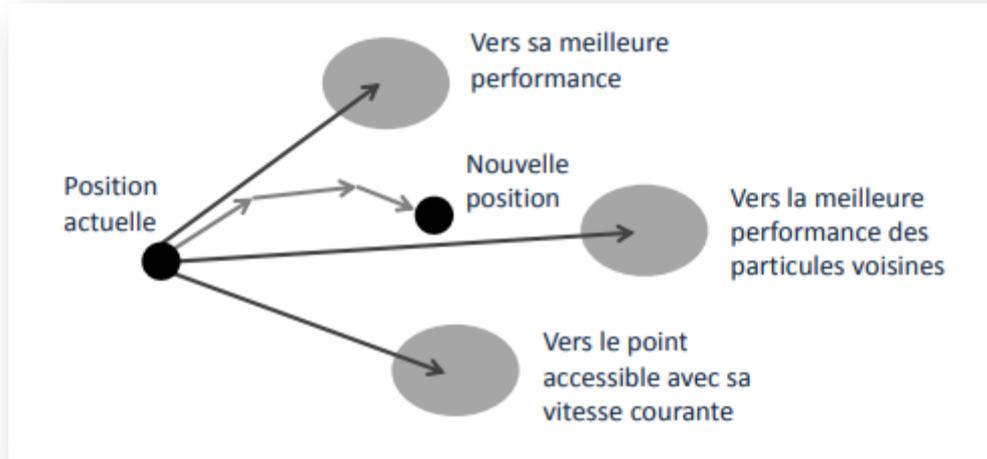


Figure 1.1: Stratégie de déplacement d'une particule. [3]

## 5.2 Formalisation

Dans un espace de recherche de dimension  $D$ , la particule  $i$  de l'essaim est modélisée par son vecteur position  $\vec{x} = (x_{i1}, x_{i2}, \dots, x_{iD})$  et par son vecteur vitesse  $\vec{v}_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ . La qualité de sa position est déterminée par la valeur de la fonction objectif en ce point. Cette particule garde en mémoire la meilleure position par laquelle elle est déjà passée, que l'on note  $\vec{P}best_i = (pbest_{i1}, pbest_{i2}, \dots, pbest_{iD})$ . La meilleure position atteinte par les particules de l'essaim est notée  $\vec{G}best = (gbest_1, gbest_2, \dots, gbest_D)$ . Nous nous référons à la version globale de PSO, où toutes les particules de l'essaim sont considérées comme voisines de la particule  $i$ , d'où la notation  $\vec{G}best$  (global best). [3]

Au départ de l'algorithme, les particules de l'essaim sont initialisées de manière aléatoire/régulière dans l'espace de recherche du problème. Ensuite, à chaque itération, chaque particule se déplace, en combinant linéairement les trois composantes citées : d'inertie, cognitive et sociale. En effet, à l'itération  $t + 1$ , le vecteur vitesse et le vecteur position sont calculés à partir de l'équation (1.1) et de l'équation (1.2), respectivement.

$$v_{i,j}^{t+1} = \omega v_{i,j}^t + c_1 r_{1,j}^t [pbest_{i,j}^t - x_{i,j}^t] + c_2 r_{2,j}^t [gbest_j^t - x_{i,j}^t], j \in \{1, 2, \dots, D\} \quad (1.1)$$

$$x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1}, j \in \{1, 2, \dots, D\} \quad (1.2)$$

où  $\omega$  est une constante, appelée coefficient d'inertie;  $c_1$  et  $c_2$  sont deux constantes, appelées coefficients d'accélération ;  $r_1$  et  $r_2$  sont deux nombres aléatoires tirés uniformément dans  $[0, 1]$ , à chaque itération  $t$  et pour chaque dimension  $j$ . Les trois composantes mentionnées ci-

dessus (i.e. d'inertie, cognitive et sociale) sont représentées dans l'équation (1.1) par les termes suivants [3]:

$\omega v_{i,j}^t$  correspond à la composante d'inertie du déplacement, où le paramètre  $\omega$  contrôle l'influence de la direction de déplacement sur le déplacement futur ;

$c_1 r_{i,j}^t [pbest_{i,j}^t - x_{i,j}^t]$  correspond à la composante cognitive du déplacement, où le paramètre  $c_1$  contrôle le comportement cognitif de la particule ;

$c_2 r_{2i,j}^t [gbest_j^t - x_{i,j}^t]$  correspond à la composante sociale du déplacement, où le paramètre  $c_2$  contrôle l'aptitude sociale de la particule.

Une fois le déplacement des particules effectué, les nouvelles positions sont évaluées et les deux vecteurs  $\vec{P}best_i$  et  $\vec{G}best$  sont mis à jour pour l'itération  $t + 1$ , suivant les deux équations (1.3) et (1.4), dans une version globale de PSO. Cette procédure est présentée dans l'Algorithme 1, où  $N$  est le nombre de particules de l'essaim. (3)

$$\vec{P}best_i(t+1) = \begin{cases} \vec{P}best_i(t), & \text{si } f(\vec{x}_i(t+1)) \geq \vec{P}best_i(t) \\ \vec{x}_i(t+1), & \text{sinon} \end{cases} \quad (1.3)$$

$$\vec{G}best(t+1) = \arg \min_{\vec{P}best_i} f(\vec{P}best_i(t+1)), 1 \leq i \leq N \quad (1.4)$$

### Algorithme 1 : Optimisation par essaims particulaires

Initialiser aléatoirement  $N$  particules : position et vitesse.

Evaluer les positions des particules

Pour chaque particule  $i$ ,  $\vec{P}best_i = \vec{x}_i$ .

Calculer  $\vec{G}best$  selon (1.4).

**Tant que** le critère d'arrêt n'est pas satisfait faire

Déplacer les particules selon (1.1) et (1.2)

Evaluer la fonction fitness des particules

Mettre à jour  $\vec{P}best_i$  et  $\vec{G}best$  selon (1.3) et (1.4)

**Fin**

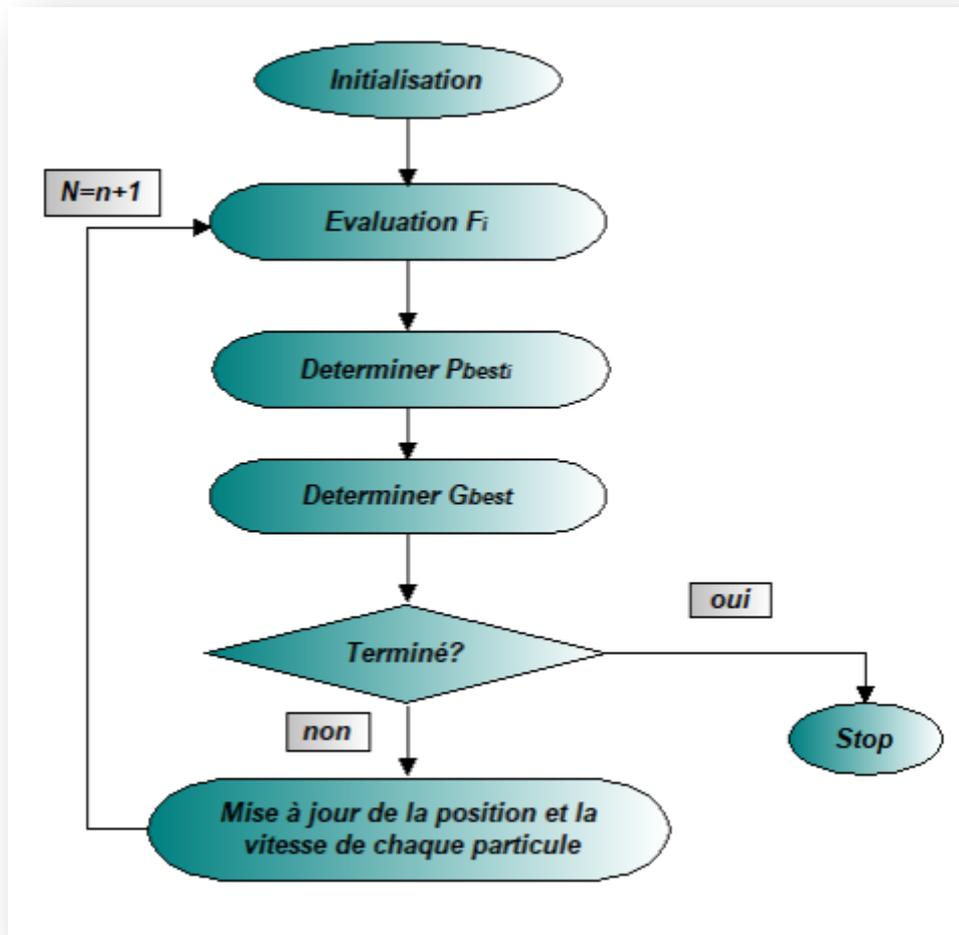


Figure 1.2: Organigramme de la méthode PSO. [2]

### 5.3 Amélioration de PSO

Malgré la simplicité et la facilité de l'utilisation de l'algorithme PSO il présente un risque de divergence ou de convergence rapide qui permet de stagner dans un optimum local, de ce fait plusieurs améliorations ont été apportées à l'algorithme de base.

#### 5.3.1 Confinement des particules

Pour éviter que le déplacement des particules soit trop rapide (l'inertie  $V$  de valeur importante), ce qui conduit à sortir de l'espace de recherche, nous pouvons introduire un nouveau paramètre  $V_{max}$ , qui permet de limiter la vitesse sur chaque dimension et ainsi de contrôler les particules.

Notons que cela ne restreint pas les valeurs de  $x_i$ , à l'intervalle  $[V_{i_{min}}, V_{i_{max}}]$ , mais limite seulement la distance maximale qu'une particule va parcourir au cours d'une itération. Cette méthode permet de contrôler la divergence de l'algorithme et de réaliser ainsi un compromis efficace entre intensification et diversification.

De plus, une stratégie supplémentaire de confinement des particules peut être introduite. Une telle stratégie permet de ramener une particule sortie de l'espace de recherche à l'intérieur de celui-ci. Dans ce cadre, plusieurs méthodes peuvent être employées : [4]

- La particule est laissée à l'extérieur de l'espace de recherche, mais on n'évalue pas sa fonction objective. Ainsi, elle ne pourra pas attirer les autres particules en dehors de l'espace de recherche.
- La particule est arrêtée à la frontière et les composantes associées à sa vitesse sont annulées
- La particule rebondit sur la frontière, elle est stoppée à la frontière mais les composantes correspondantes de la vitesse sont multipliées par un coefficient tiré aléatoirement dans l'intervalle  $[-1,0]$ . [4]

### 5.3.2 Coefficient de constriction

L'étude de la dynamique des particules au sein de l'essaim a conduit à la recherche de solutions pour éviter la divergence de l'algorithme, comme par exemple l'introduction du paramètre  $V_{max}$  que nous avons vu dans le paragraphe précédent et qui permet de limiter la divergence des particules. De nombreuses études ont été menées sur la dynamique des particules concernant l'analyse des conditions de convergence de l'essaim.

La combinaison des paramètres  $\omega$ ,  $c_1$  et  $c_2$  permet de régler l'équilibre entre les phases de diversification et d'intensification du processus de recherche. L'utilisation d'un coefficient de constriction  $\chi$  (ou facteur de constriction) permet de mieux contrôler la divergence de l'essaim et de s'affranchir de la définition de  $V_{max}$ . Cette variante de PSO, qui a été largement utilisée dans la littérature, est connue sous le nom de canonical PSO. En utilisant ce coefficient, l'équation (1.3.1) devient [4]:

$$v_{i,j}(t+1) = \chi(v_{i,j}(t) + \phi_1 r_1 (pbest_{i,j}(t) - x_{i,j}(t)) + \phi_2 r_2 (gbest_{i,j}(t) - x_{i,j}(t))) \quad (1.5)$$

Avec :

$$\chi = \frac{2}{\phi - 2 + \sqrt{\phi^2 - 4\phi}} \quad (1.6)$$

Où :  $\phi = \phi_1 + \phi_2, \phi > 4$ .

### 5.3.3 Coefficient d'inertie

Le coefficient d'inertie  $\omega$ , introduit par Y.Shi et Eberhart, contrôle l'influence de la direction de la particule sur le déplacement futur. Le but de l'introduction de ce paramètre est de réaliser un équilibre entre la recherche locale (exploitation) et la recherche globale (exploration). L'intensité de l'exploration de l'espace de recherche dépend de la valeur du poids d'inertie, une grande valeur de  $\omega$  facilite l'exploration globale, alors qu'une petite valeur facilite l'exploration locale. Du fait de son influence sur les performances de l'algorithme PSO, le poids d'inertie a suscité un grand intérêt de la part de la communauté des chercheurs

. Le coefficient d'inertie  $\omega$  varie linéairement avec le temps selon la formule suivante :

$$\omega = \omega_{min} + (w_{max} - \omega_{min}) \frac{iter}{iter_{max}} \quad (1.7)$$

Où :  $iter$  est l'itération courante et  $iter_{max}$  est le nombre maximal d'itérations. Les coefficients max et min désignent respectivement les valeurs maximum et minimum du coefficient. Dans une autre variante Y.Shi et Eberhart ont proposé une valeur du coefficient d'inertie choisie au hasard selon une distribution uniforme telle que  $\omega \in [0.5 \ 1]$ . [4]

### 5.3.4 Topologie de voisinage :

La topologie du voisinage constitue la structure du réseau social et définit avec qui chacune des particules va pouvoir communiquer. Il existe de nombreuses combinaisons dont les suivantes sont les plus utilisées :

**Topologie en étoile :** chaque particule est reliée à toutes les autres, l'optimum du voisinage est l'optimum global.

**Topologie en anneau :** chaque particule est reliée à  $n$  particules ( $n = 3$  en général), c'est la plus utilisée.

**Topologie en rayon :** les particules ne communiquent qu'avec une seule particule centrale.[4]

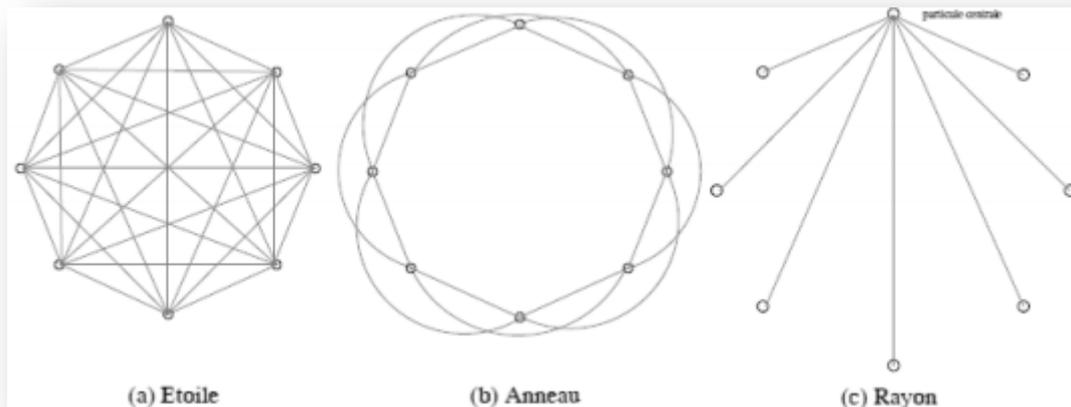


Figure 1.3: Topologies de voisinage. [4]

## 6. Optimisation par trois essaims particulaires coopératives

### 6.1 Principe générale

L'optimisation par trois essaims particulaires coopératifs ou Three Swarms Cooperative-Particle Swarm Optimization (TSC-PSO), est une méthode basée sur PSO et qui conçoit trois essaims séparés opérant sur le même espace de recherche. Deux essaims appelées « les excellents essaims » volent vers la particule optimale tandis que le dernier, appelé « le mauvais essaim » s'envole loin de la particule optimale.

L'envol des particules des excellents essaims est affecté par l'expérience de particule optimale dans ses essaims. Dans le mauvais essaim, la particule est affectée par l'historique d'envol de cet essaim uniquement.

Si la valeur de la particule optimale dans tous les essaims de l'ensemble du processus, ne s'améliore pas après plusieurs générations et l'emplacement optimal n'est pas trouvé au cours de la recherche, alors un excellent essaim peut être choisi arbitrairement et toutes les particules de l'excellent essaim échangent avec celles du mauvais essaim. Cet échange permet de poursuivre la recherche de la particule optimale dans l'excellent essaim et mauvais essaim. La particule optimale de l'excellent l'essaim devrait échanger avec celle du mauvais essaim au cours de leur échange. Mais l'extremum d'un essaim séparé est invariable ce qui conserve la même valeur qu'avant.

Ainsi, pendant tout le cours de recherche, la particule d'un excellent essaim peut changer de mode de vol et ne pas stagner en raison de la rencontre d'un extremum local, ce qui peut réduire la possibilité que l'essaim de particules de l'ensemble du processus converge vers l'extremum local.

L'équation de vitesse et de position optimale de particule dans un excellent essaim est définie comme suit :

$$v_{id}(t+1) = \omega \cdot v_{id}(t) + c_1 r_1 (p_{id}(t) - x_{id}(t)) + c_2 \cdot r_2 (p_{gd}(t) - x_{id}(t)) + c_3 \cdot r_3 (p_{pd}(t) - x_{id}(t)) + c_4 \cdot r_4 (p_{qd}(t) - x_{id}(t)) \quad (1.8)$$

Et

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (1.9)$$

Où :

$x_i$  est la position de de  $i$ ème particule ( $i=1,2,\dots ,n$ ).

$v_i$  La vitesse correspond au particule  $i$ .

$d$  La dimension de particule.

$\omega$  Coefficient d'inertie.

$c_1, c_2, c_3$  et  $c_4$  Les coefficients d'accélération.

$r_1, r_2, r_3$  et  $r_4$  Des nombres aléatoires tirés uniformément et indépendamment dans  $[0, 1]$ .

$p_{id}$  La position optimale historique individuelle de particule  $i$ .

$p_{gd}$  La position optimale de particule dans l'ensemble du processus, est également la meilleur parmi toutes les positions optimales des essais séparés.

$p_{pd}$  La position optimale de particule de son excellent essaim.

$p_{qd}$  La position optimale de particule de l'autre excellent essaim.

La vitesse et la position optimale de particule dans la deuxième dimension ( $1 \leq d \leq D$ ) en mauvais essaim est défini comme suit :

$$v_{id}(t+1) = \omega \cdot v_{id}(t) + c_1 r_1 (p_{id}(t) - x_{id}(t)) + c_2 \cdot r_2 (p_{gd}(t) - x_{id}(t)) + c_3 \cdot r_3 (p_{sd}(t) - x_{id}(t)) \quad (1.10)$$

$$x_{id}(t + 1) = x_{id}(t) + v_{id}(t + 1) \quad (1.11)$$

Où  $p_{sd}$  La position optimale de particule dans le mauvais essaim.[5]

### 6.2 L'approche coopérative des trois essais

Détaillé comme suit :

**Etape 1 :** définir les variables de stratégie, et le/les paramètre(s) qui doivent être optimisé, la plage de recherche est déterminée en fonction du problème concret.

**Etape 2 :** La valeur appropriée  $f(X_i)$  de la particule  $X_i$  de 3 essais distincts est évaluée. La coordonnée de chaque particule dans un essaim excellent et mauvais défini l'emplacement actuel. Si l'emplacement actuel est meilleur que l'emplacement optimal historique pour la particule, l'emplacement historique optimal de la particule est substitué par l'emplacement courant de celui-ci.

**Etape 3 :** Si l'emplacement optimal historique individuel de certaines particules en excellent ou mauvais essaim est meilleur que le courant extremum de l'essaim, le numéro de séquence de la particule est enregistré et l'extremum de l'essaim est mis à jour. Si l'extremum actuel dans l'ensemble du processus est meilleur que la valeur historique optimale individuelle pour certaine particule dans l'ensemble du processus, le numéro de séquence de la particule est enregistrée et l'extremum dans l'ensemble du processus a besoin de mettre à jour.

**Etape 4 :** M particules sont choisies arbitrairement en dimension D espace de recherche d'essaim aléatoire, avec l'expression de  $X_{ij} = (x_{ij1}, x_{ij2}, \dots, x_{ijD})$ ,

$i = 1, 2, \dots, m$ ,  $j = p, q, s$ . La position optimale que chaque particule a recherché  $p_{ij} = (p_{ij1}, p_{ij2}, \dots, p_{ijD})$ , La position optimale que tous les particules ont recherché de loin  $p_{gi} = (p_{gj1}, p_{gj2}, \dots, p_{gjD})$ , la position actuel que chaque particule a recherché dans l'espace de recherche de dimension D  $X_{ij} = (x_{ij1}, x_{ij2}, \dots, x_{ijD})$ , et la vitesse de vol actuelle de chaque particule  $v_{ij} = (v_{ij1}, v_{ij2}, \dots, v_{ijD})$  sont tout enregistrées.

**Etape 5 :** La nouvelle vitesse de particule dans un excellent essaim est calculée selon (1.8). Le nouvel emplacement de la particule dans l'excellent essaim est calculé selon (1.9). La nouvelle vitesse de particule dans le mauvais essaim est calculée selon (1.10). La nouvelle position de la particule dans le mauvais essaim est calculée en fonction à (1.11).

Si le prochain emplacement que chaque particule génère à travers la recherche n'est pas dans la zone faisable, la position de la particule est ramenée à la zone.

**Etape 6 :** L'extremum dans l'ensemble du processus est jugé dans la Nième étape. Si l'extremum ne s'améliore, un essaim est choisi parmi les deux excellents essaims et échangent entièrement avec les particules dans le mauvais essaim ; la particule optimale dans un excellent essaim échange avec la particule optimale en mauvais essaim ; les extremums de deux essaims séparés échangent également en conséquence. Sinon, il continue à suivre les étapes.

**Etape 8 :** Le nombre itératif est également évalué. Si le nombre est plus grand que le nombre itératif maximal, il est temps d'arrêter de rechercher et écrire l'extremum et la position optimale dans le processus complet. Sinon, il faut passer à l'étape 2 pour continuer la recherche. [5]

## 7. Conclusion

Parmi les méthodes et les algorithmes utilisés pour résoudre des problèmes, un intérêt particulier a été porté à la méthode d'optimisation par essaim de particules (PSO). Cette méthode, qui est inspirée du comportement social des animaux évoluant en essaim, a rencontré un succès remarquable depuis sa création, grâce à sa simplicité. Elle présente l'avantage d'être efficace sur une vaste gamme de problèmes. Cependant, PSO présente un problème majeur, qui rebute encore certains utilisateurs : le problème de la convergence rapide, qui peut conduire les algorithmes de ce type à stagner dans un optimum local. De nombreux travaux ont été proposés pour améliorer les performances de PSO, tout en essayant de remédier à ce problème. [5]

# Chapitre 2 : Généralités sur la modélisation et le contrôle du trafic routier

---

## 1. Introduction

La croissance permanente de la population dans les villes modernes a augmenté le nombre des véhicules. Ainsi, le problème de la congestion du trafic est devenu l'un des principaux problèmes à résoudre par les systèmes de régulation du trafic d'aujourd'hui. Particulièrement, au cours de la dernière décennie, la congestion du trafic a attiré une attention significative à cause de la crise énergétique mondiale et les préoccupations environnementales. Le trafic routier est un phénomène complexe d'une part en raison du nombre élevé d'acteurs qui y participent, d'autre part à cause du caractère très maillé du réseau sur lequel il se déroule. Même s'il existe de bons systèmes de gestion de la route et des infrastructures routières suffisantes pour le transport, nous faisons encore face à une congestion du trafic, en particulier, aux carrefours à feux.[6]

## 2. Les variables de trafic

### 2.1 Véhicules considéré individuellement

#### 2.1.1 Notion de trajectoire

On s'intéresse à un même véhicule, supposons que toutes les dix secondes on puisse mesurer sa position avec une précision suffisante (par exemple grâce à un système de type GPS embarqué), on pourra alors schématiser l'évolution du véhicule sur la voie par la figure ci-dessous :[7]

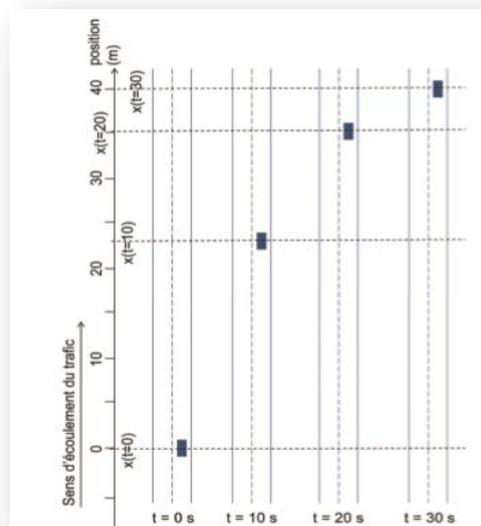


Figure 2.1: position sur la voie d'un véhicule à quatre instants différents. [7]

#### 2.1.2 Mesures de la position longitudinale

On s'intéresse uniquement à la position longitudinale  $\mathbf{x}(t)$  par rapport à la voie. Ceci est lié à l'appareil de mesure qui, historiquement, est un capteur de tours de roues. On résume donc la voie à une droite, le long de laquelle on mesure la position. On parle donc couramment de trajectoire dans un plan  $(\mathbf{x}, t)$ . On peut représenter ces mesures sur la figure suivante. Remarquons que le véhicule étudié est resté immobile entre les instants  $t = 22$  s et  $t = 24$  s. On peut déduire de l'information de cette figure la distance parcourue (environ 3 mètres) par le véhicule pendant une seconde entre  $t=4$ s et  $t=5$  s par exemple. [7]

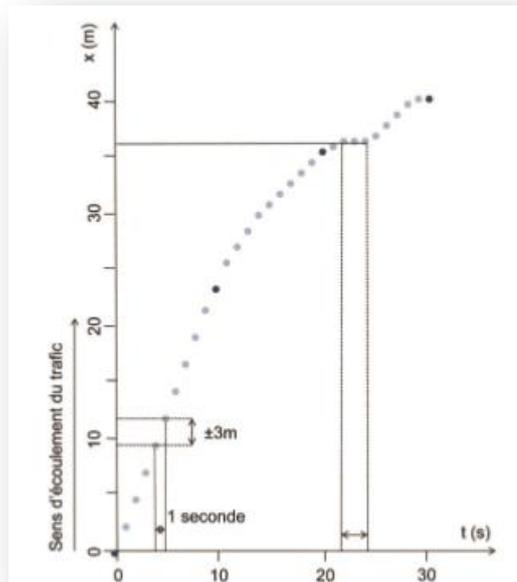


Figure 2.2: trajectoire d'un véhicule dans le plan  $(x, t)$ . [7]

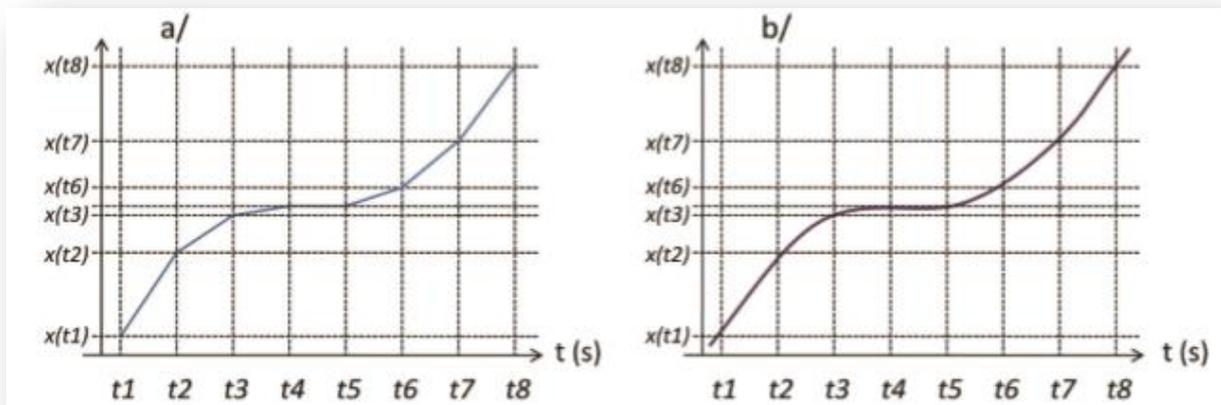


Figure 2.3: deux trajectoires de véhicules : à gauche (a/) une trajectoire interpolée entre deux mesures de position successives ; à droite (b/) à partir d'une mesure continue de la vitesse et de la position. [7]

### 2.1.3 Mesures de $t(x)$

Si, au lieu de disposer d'une mesure embarquée de la position du véhicule à chaque pas de temps, on place le long de la route des observateurs régulièrement espacés, chargés de mesurer les dates de passage du véhicule qui nous intéresse, on associera les instants aux points de mesure. Ceci est représenté sur la figure suivante. [7]

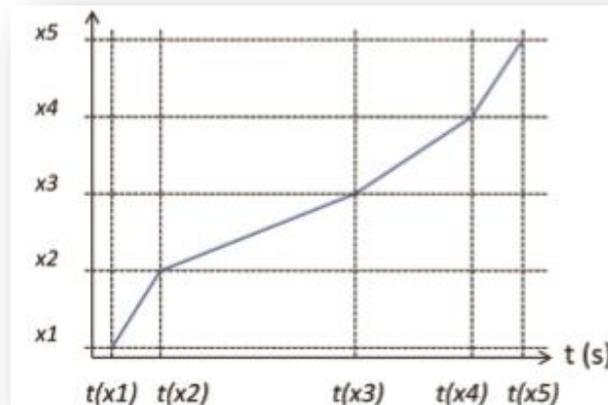


Figure 2.4: évolution d'un véhicule dont la position est mesurée en des points régulièrement espacés de la voie. [7]

## 2.1.4 Variables et caractéristiques d'un véhicule

**La position** d'un véhicule mesurée à un instant est notée  $x(t)$ .

**La vitesse** d'un véhicule est la dérivée de la position par rapport au temps :

$$V(t) = \dot{x}(t) = \frac{dx(t)}{dt}$$

**L'accélération** est la dérivée de la vitesse d'un véhicule par rapport au temps et donc la dérivée seconde de la position par rapport au temps :

$$a(t) = \ddot{x}(t) = \frac{d^2x(t)}{dt^2} = \frac{dV(t)}{dt}$$

**La trajectoire** d'un véhicule est l'ensemble des points associés à la date de passage d'un véhicule. Elle peut être représentée dans un plan  $(x, t)$ , soit à partir de mesures à chaque pas de temps de la position, soit à partir de mesures des instants de passage en des points régulièrement espacés de la route. [7]

## 2.1.5 Temps de parcours individuel

Le temps de parcours individuel est la mesure du temps passé par un véhicule pour aller d'un point à un point. C'est donc l'écart entre la date d'entrée sur le tronçon (A) et la date de sortie du tronçon (B). Il est connu à l'instant  $t = T_{\text{passage}}(B)$  où le véhicule passe au point B de sortie :

$$T_{P_{A \rightarrow B}}(t) = T_{\text{passage}}(B) - T_{\text{passage}}(A)$$

Du temps de parcours individuel entre deux points, on peut déduire la vitesse moyenne individuelle sur ce tronçon ( $L_{A \rightarrow B}$  étant la distance entre l'entrée et la sortie du tronçon) : [7]

$$V_{A \rightarrow B}(t) = L_{A \rightarrow B} / T_{P_{A \rightarrow B}}(t)$$

### 2.2 Variables concernant deux véhicules successifs

Lorsqu'un véhicule suit un autre véhicule, on parle de véhicule suiveur et de véhicule leader ou de véhicule suivi. On peut les observer également de deux manières complémentaires l'une de l'autre :

– en un instant  $t$  ; on mesure la distance entre ces deux véhicules. Ici peut intervenir la longueur du véhicule. Ceci conduit soit à l'espacement ou à la distance inter-véhiculaire (qui est la distance entre deux avants ou arrières de véhicules et qui se dit en anglais « spacing »), soit au créneau (en anglais : « space gap »), qui est la distance séparant l'arrière d'un véhicule de l'avant du véhicule qui le suit ;

– en un point  $x$  ; dans ce cas, on mesure le temps séparant les passages de deux véhicules successifs. De la même manière on peut mesurer le temps séparant le passage de l'avant ou de l'arrière de deux véhicules successifs : le temps inter-véhiculaire, ou le temps séparant le passage de l'arrière d'un véhicule du passage de l'avant du véhicule suivant. Ce dernier est appelé intervalle, créneau temporel ou « time gap » en anglais. [7]

Ces notions sont illustrées sur la figure suivante :

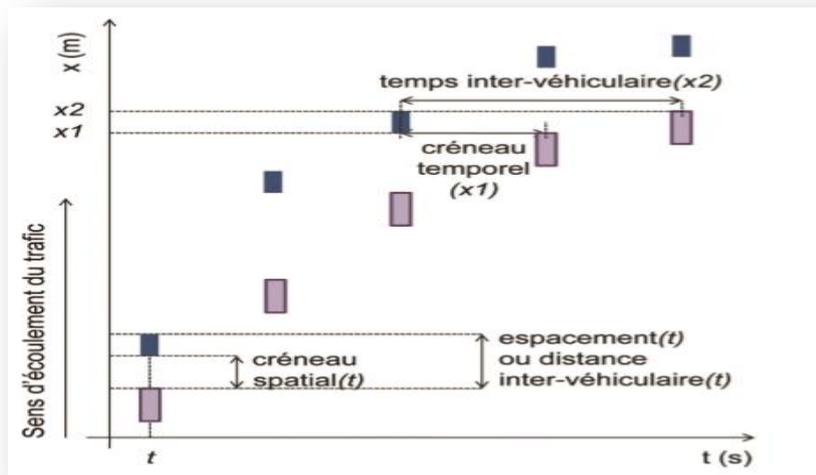


Figure 2.5: trajectoires de deux véhicules : un véhicule leader et son véhicule suiveur. [7]

## 2.3 Variables Concernant un flot de véhicules

Dans la suite on ne considérera qu'un sens d'écoulement. Un flot de véhicules est l'ensemble des véhicules parcourant une voie à une période donnée. Il est composé de véhicules divers par leurs caractéristiques physiques (poids, longueur, puissance...) et autres (vitesse désirée, nombre de passagers...).

Pour mesurer les flots de véhicules, on s'intéresse à des variables globales dont les définitions sont données ci-après :

**Le débit Q** : est le nombre  $N$  de véhicules passant pendant une période  $\Delta T$  en un point  $x$ , rapporté à la durée de la période :  $Q_{\Delta T}(x) = Q(x, t \rightarrow t + \Delta t) = \frac{N}{\Delta t}$  . Il est exprimé en nombre de véhicules par unité de temps véh/h ou véh/s(ou dans le cas de trafic hétérogène PCE/h ou PCE/s) (Figure 6)

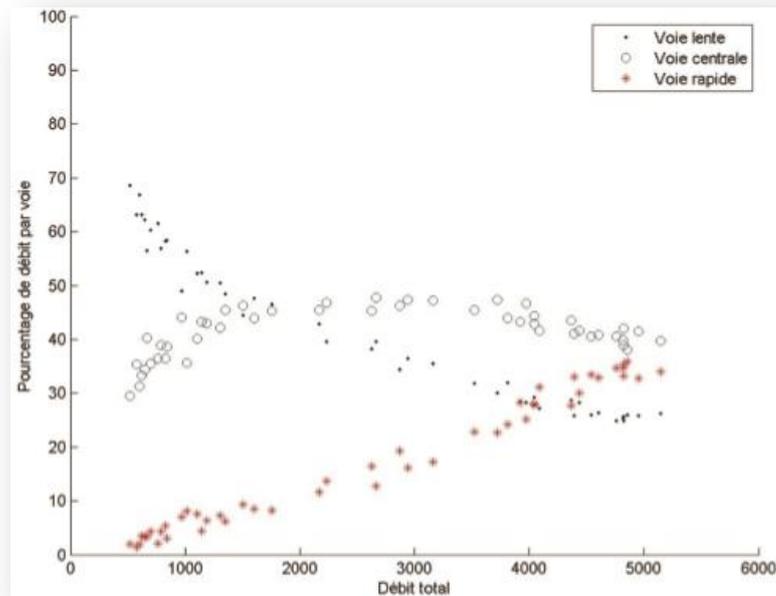


Figure 2.6: évolution de la proportion du débit sur chaque voie en fonction du débit total (en véh/h). [7]

**La densité K** : Ou concentration, le nombre  $N$  de véhicules entre deux points  $x_1$  et  $x_2$  à un instant  $t$  :  $K_{\Delta x}(t) = \frac{N}{\Delta x}$ ,  $\Delta x = x_1 - x_2$ . Cette variable est exprimée en véh/Km (ou PCE/Km) (Fig. 7)

**Vitesse moyenne temporelle** : C'est la moyenne des vitesses de  $N$  véhicules passant entre deux point  $x_1$  et  $x_2$  : [8]

$$V_{\Delta x}(t) = \frac{1}{N} \sum_{j=1}^N V_{\Delta x}^j(t), \Delta x = x_1 - x_2$$

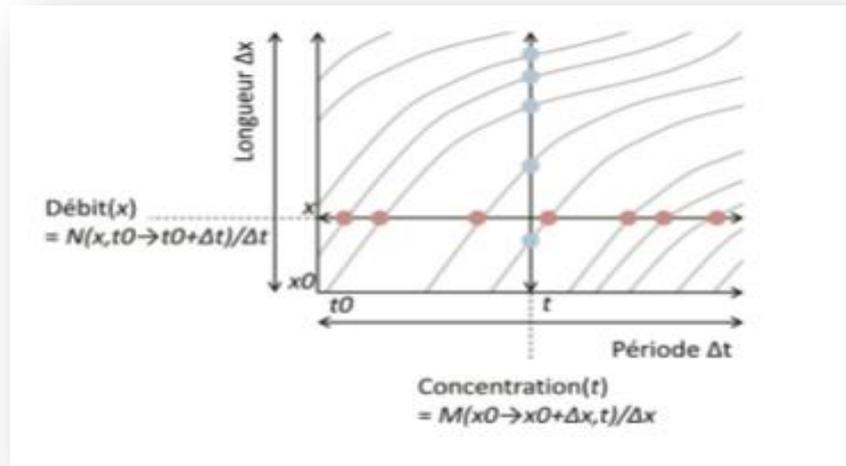


Figure 2.7: Représentation du débit  $Q$  et concentration  $K$  par rapport aux trajets de véhicules dans une route. [8]

**Vitesse moyenne spatiale :** C'est la moyenne des vitesses de  $N$  véhicules passant à un point  $x$  pendant une période  $\Delta t$  donnée : [6]

$$V_{\Delta x}(t) = \frac{1}{N} \sum_{j=1}^N V_{\Delta t}^j(x)$$

### Diagramme fondamental du trafic

Le diagramme fondamental du trafic est une alternative aux modélisations précédentes et permet de décrire la relation entre le débit et la densité d'un réseau routier [CSS00]. Il est observé théoriquement, et plusieurs variations sont identifiables expérimentalement [Ker00]. Il distingue deux états de trafic différents : le cas où le trafic est fluide (c.-à-d., vitesse proche de la vitesse libre) et le cas où il est embouteillé (c.-à-d., les véhicules se déplacent à une vitesse inférieure à la vitesse libre). La figure 9 (a) représente la forme générale de ce diagramme. Des variantes existent, telles que celle suggérée par la théorie des trois phases du trafic, de Kerner [Ker00]. Elle considère un état supplémentaire d'embouteillage, comme représenté sur la figure 9 (b). [7]

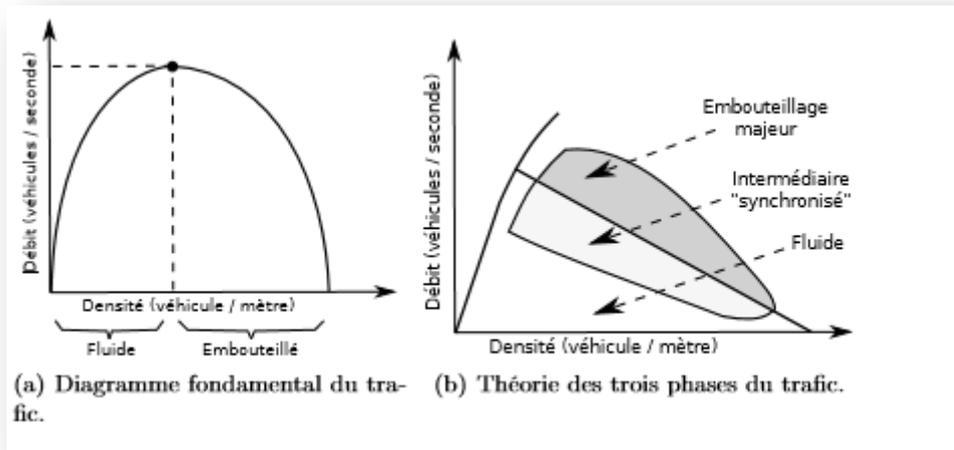


Figure 2.8: Diagrammes débit-densité .[7]

### 3. Modélisations du trafic routier

De nos jours le trafic routier et les systèmes de transports jouent un rôle primordial dans le développement des pays et des sociétés en plus, les études faites sur le trafic routier deviennent très indispensable à cause de l'augmentation croissante des besoins en déplacement des biens et des personnes. Ainsi, beaucoup de travaux ont été effectués sur le trafic routier dans le but de l'optimisation, de l'utilisation infrastructures de transport, l'amélioration de la sécurité routière et de la sureté, etc. [9]

Depuis longtemps, différentes approches ont été utilisées et plusieurs outils et technologies ont été exploités dans le cadre de l'étude et de la modélisation du trafic routier. Plusieurs modèles ont été mis en œuvre parmi lesquels on compte 3 types de modèles : [9]

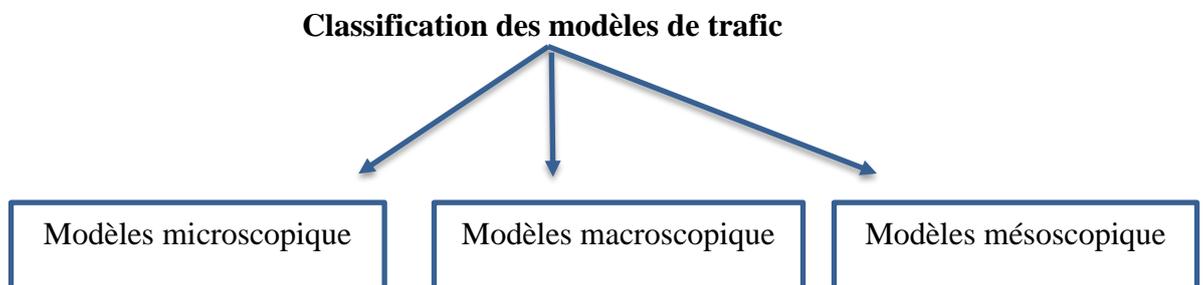


Figure 2.9: classification des modèles de trafic.

### 3.1 Modèles microscopiques :

Un modèle microscopique décrit à la fois le comportement spatio-temporel des composants du système (véhicules et conducteurs) et leurs interactions à un niveau élevé de détail (individuellement). Par exemple, pour chaque véhicule participant, le changement de voie de circulation est décrit comme une chaîne de décisions du conducteur.

De la même façon que les modèles microscopiques, les modèles microscopiques décrivent les caractéristiques de chaque véhicule participant à la circulation. Cependant, en plus de la description détaillée du comportement en trafic, le comportement de contrôle du véhicule (par exemple, le changement des vitesses) est modélisé aussi selon les conditions environnementales. En plus, même le fonctionnement des parties spécifiques du véhicule (sous-ensembles) est décrit.

Les modèles microscopiques sont appropriés pour la modélisation des interactions entre véhicules à un haut niveau de détail nécessaire pour évaluer plusieurs Systèmes de Transport Intelligents, mais ils sont limités à de petites zones en raison de la grande quantité de données nécessaires.

### 3.2 Modèles macroscopiques

Les modèles macroscopiques décrivent le trafic à un niveau élevé d'agrégation, comme un flux de véhicules, sans égard aux parties composantes. Ces modèles sont généralement utilisés pour des opérations de planification et de contrôle, portant sur des grands réseaux et de longues périodes de temps. Le trafic routier est représenté d'une façon compacte en utilisant une série de variables interdépendantes comme le flux, la densité et la vitesse. Les manœuvres caractéristiques d'un véhicule, telles que le changement de voies, ne peuvent pas être représentées.

Il y a une analogie intéressante entre la description mathématique de la circulation routière et l'écoulement d'un fluide, analogie exploitée par ces modèles. Cependant, il y a des phénomènes spécifiques au trafic qui ne correspondent pas à l'écoulement d'un fluide, comme l'instabilité, la congestion routière ou le trafic de type arrêt-et-aller (« stop-and-go »). [9] Toutefois les modèles macroscopiques représentent des outils importants pour la simulation, la prévision, l'estimation et la conception de stratégies de contrôle. [9]

### 3.3 Modèles mésoscopiques

Un modèle mésoscopique ne différencie pas et ne suit pas les véhicules comme entités individuelles, mais il spécifie le comportement de véhicules en termes probabilistes. Ainsi, le trafic est représenté par de petits groupes de véhicules, pour lesquels les activités et les interactions sont décrites à un faible niveau de détail. Par exemple, une manœuvre de changement de voie peut être représentée pour un véhicule comme un événement instant, où la décision de faire un changement de voie est basée sur les densités relatives des bandes et les différences de vitesse. Certains modèles mésoscopiques sont obtenus par analogie avec la théorie cinétique de gaz, en décrivant la dynamique des distributions de vitesse. [9]

## 4. Contrôle de feux de signalisation

### 4.1 Objectifs et domaine d'emploi

L'emploi des feux de circulation a pour but d'assurer la sécurité de toutes les usagères et tous les usagers de la voirie, piétons et conducteurs, et de faciliter l'écoulement des flux de circulation denses. On peut citer comme exemples d'emploi :

- La gestion de la circulation aux intersections ;
- La traversée des piétons, autour des intersections gérées par des feux et les moments où la circulation est plus intense ou lorsque le sentiment d'insécurité des piétons important ;
- L'exploitation par sens uniques alternés d'une section où le croisement est impossible ou dangereux (ouvrage d'art étroit, emprise de travaux, etc.) ;
- L'affectation de certaines voies d'une chaussée à un sens de circulation en fonction des besoins, ou leur condamnation momentanée ;
- La gestion d'un point de contrôle des personnes ou des véhicules nécessitant leur arrêt (péage) ;
- La protection d'obstacles intermittents (passages à niveau, traversées de voies de tramways, ponts mobiles, passages d'avions, avalanches, etc.).[10]

### 4.2 Principe

Lorsque le volume de la circulation excède la capacité d'un carrefour, il faut songer à installer des feux. Les critères qui gouvernent l'installation de feux sont basés sur :

- Le volume des voitures et des piétons traversant un carrefour ;
- Les retards subis par les véhicules et les piétons en provenance des rues transversales ;

- Le nombre de collisions à ce carrefour.

Un système de feux bien ajusté peut augmenter considérablement le nombre de véhicules traversant un carrefour ou améliorer la sécurité des piétons et des voitures.

Les feux sont généralement déclinés à partir de deux couleurs de base : le rouge pour fermer, le vert (ou encore le bleu plus rarement) pour ouvrir. Le jaune-orangé est également utilisé et sert à signaler le passage du feu vert au feu rouge (et aussi, dans certains pays, du feu rouge au feu vert). [10] Ceci laisse le temps à différents usagers de passer en évitant de bloquer certaines voies de circulation.

### 4.3 Stratégies de régulation

Les paramètres qui sont utilisés en régulation du trafic urbain sont au nombre de 3 : le cycle, les phases (temps de vert des différentes voies) et le décalage entre les verts de carrefours adjacents.

- **Le cycle d'un carrefour** : est le temps que met le carrefour pour se retrouver dans sa configuration de départ, chaque feu ayant eu son temps de vert.

- **Les phases** : sont les états (vert, rouge, orange, orange clignotant) dans lesquels se trouvent les feux ; chaque feu change plusieurs fois de phases durant un cycle.

- **Le décalage** : exprime le temps que met un carrefour pour se mettre au vert sur la voie principale lorsque le carrefour amont s'est mis au vert sur la même voie ; c'est par ce paramètre qu'il est possible d'agir sur les coordinations d'axe et de donner aux véhicules une vitesse limite de déplacement. [11]

#### 4.3.1 Stratégie multiprogrammation (prédéterminée ou cyclique)

La multiprogrammation permet d'adapter les feux aux situations caractéristiques du trafic, répétitives et prévisibles. Il s'agit de plans de feux fixes adaptés à des moments précis de la journée qui sont changés par une horloge ou par un autre moyen de décision. En tout état de cause, il s'agit de plans de feux mémorisés dans une bibliothèque. Ce mode de fonctionnement est en général utilisé dans le cas des carrefours isolés, pendant la phase de mise au point du système ou lors de fonctionnements dégradés. [11]

#### 4.3.2 Stratégie micro-régulation (semi-adaptative ou acyclique)

La micro-régulation (action sur un seul carrefour) permet de s'adapter aux variations aléatoires et imprévisibles du trafic [12]. Ce principe utilise des informations données par des

capteurs et n'a de sens que si ceux-ci bénéficient d'une maintenance appropriée. Cette technique est mise en œuvre sur des carrefours isolés ou sur des carrefours s'intégrant dans une stratégie globale de régulation. Elle a des possibilités d'anticipation-prolongation de vert pour les autobus ou les véhicules prioritaires. Dans le cas des carrefours isolés, les boucles sont gérées et utilisées directement par le contrôleur de carrefour qui décide lui-même du partage du vert entre la principale et la transversale. Dans le cas de carrefours s'intégrant dans une stratégie globale de régulation, les informations en provenance des boucles ou autres capteurs sont envoyées au coordinateur central ou au coordinateur de zone. Celui-ci applique une politique globale ou permet une adaptativité gérée localement à certains moments et dans certaines limites. [11]

### 4.3.3 Stratégie temps réel (adaptative)

Il s'agit essentiellement des stratégies qui proposent des réglages de feux en fonction des paramètres mesurés en général par des capteurs plus nombreux que dans le cas de la multiprogrammation [12]. Ces systèmes essaient de coller le plus possible à l'état réel du trafic. Il est difficile de mettre en œuvre un système temps réel [11] car:

- Les données à recueillir, donc les transmissions, sont coûteuses.
- Le temps de calcul des plans de feux est beaucoup trop long, à moins d'utiliser des modèles de trafic extrêmement simplifiés dénaturant totalement l'image de ce trafic sur un réseau important.

Il est intéressant de noter que le principe même d'une stratégie totalement adaptative ne semble pas satisfaisant car il consiste à essayer de faire correspondre au mieux plans de feux et situation réelle, or la régulation doit gérer et non suivre le trafic [11]

## 4.4 Les critères à optimiser

Typiquement, il y a trois critères utilisés fréquemment pour évaluer la performance d'une stratégie de régulation du trafic, à savoir :

- Débit de sortie (en anglais, « throughput ») ;
- Longueur moyenne de queue (en anglais, « average queue length ») ;
- Temps moyen d'attente (en anglais, « average waiting time »). [6]

### 5. Conclusion

La gestion du trafic routier est l'ensemble des dispositions destinées à répartir les flux de circulation dans le temps et dans l'espace. La modélisation du trafic routier permet la représentation des activités humaines selon le besoin en transport (de personnes ou de marchandises), elle nécessite une description mathématique de toutes les variables et des paramètres qui interviennent et qui influent sur le trafic.

# Chapitre 3 :L'outil de simulation du trafic urbain

## SUMO

---

### 1. Introduction

Pour une évaluation cohérente des performances d'un contrôle sur les réseaux véhiculaires, il est nécessaire de créer un scénario proche de la réalité basé sur un réseau routier déterminé. Plusieurs outils et programmes ont été développés pour aider à effectuer cette estimation tel que SUMO, VISSIM, AnyLogic, GLD traffic simulator . . . etc.

### 2. Simulation

La simulation est la reproduction d'un comportement d'un système réel en utilisant un modèle le décrivant. D'un point de vue informatique, la simulation est l'exécution d'un modèle pas-à-pas avec un ensemble de données d'entrée ce qui permet de suivre l'évolution d'un système et comprendre mieux son comportement et explorer les différentes stratégies. On parle d'une formalisation de simulation de modèle analytique dont les résultats ont une interprétation théorique. La simulation permet d'appliquer le modèle, l'évaluer, le calibrer et le valider pour des éventuelles utilisations qui peuvent faire objet de conclusions utiles sur un système.

Ils existe de nombreux outils pour la simulation microscopique de trafic routier, comme le persan et VISSIM . Mais un outil de simulation de trafic routier populaire parmi la communauté de recherche est la simulation de la mobilité urbaine (Simulation of Urban MObility)

### 3. Le simulateur SUMO

SUMO est l'acronyme de "Simulation of Urban MObility". C'est un progiciel de simulations de trafic routier open source sous licence ONU public (OPL), mis en œuvre en 2001, avec une première version open source en 2002. Alors qu'il existe des simulations de trafic open source disponibles, la plupart d'entre eux ont été mis en œuvre dans le cadre d'une thèse d'étudiant et n'ont pas été pris en charge par la suite. SUMO est plate-forme de

simulation commune pour soutenir la communauté de simulation de trafic avec un outil dans lequel les algorithmes et modèles peuvent être implémentés et comparés.



Figure 3.1:SUMO

### 3.1 Principales fonctionnalités

SUMO n'est pas seulement une simulation de trafic, mais plutôt une suite d'applications qui aident à préparer et à exécuter les simulations de trafic. Comme la simulation de trafic en SUMO nécessite la représentation des réseaux routiers et de la demande du trafic dans un format précis, les deux doivent être importés ou générés à l'aide de différentes sources.

Les réseaux routiers SUMO peuvent être soit générés à l'aide d'une application nommée «netgenerate» ou générée par l'importation d'une carte de route numérique. L'importateur du réseau routier «netconvert» permet de lire les réseaux d'autres simulateurs de trafic comme VISUM, Vissim ou MATsim. Il lit également d'autres formats, sous forme de fichiers de formes ou Open Street Map. Outre ces formats, «netconvert» est capable de lire des formats moins connus, comme Format réseau RoboCup ou openDRIVE. [8]

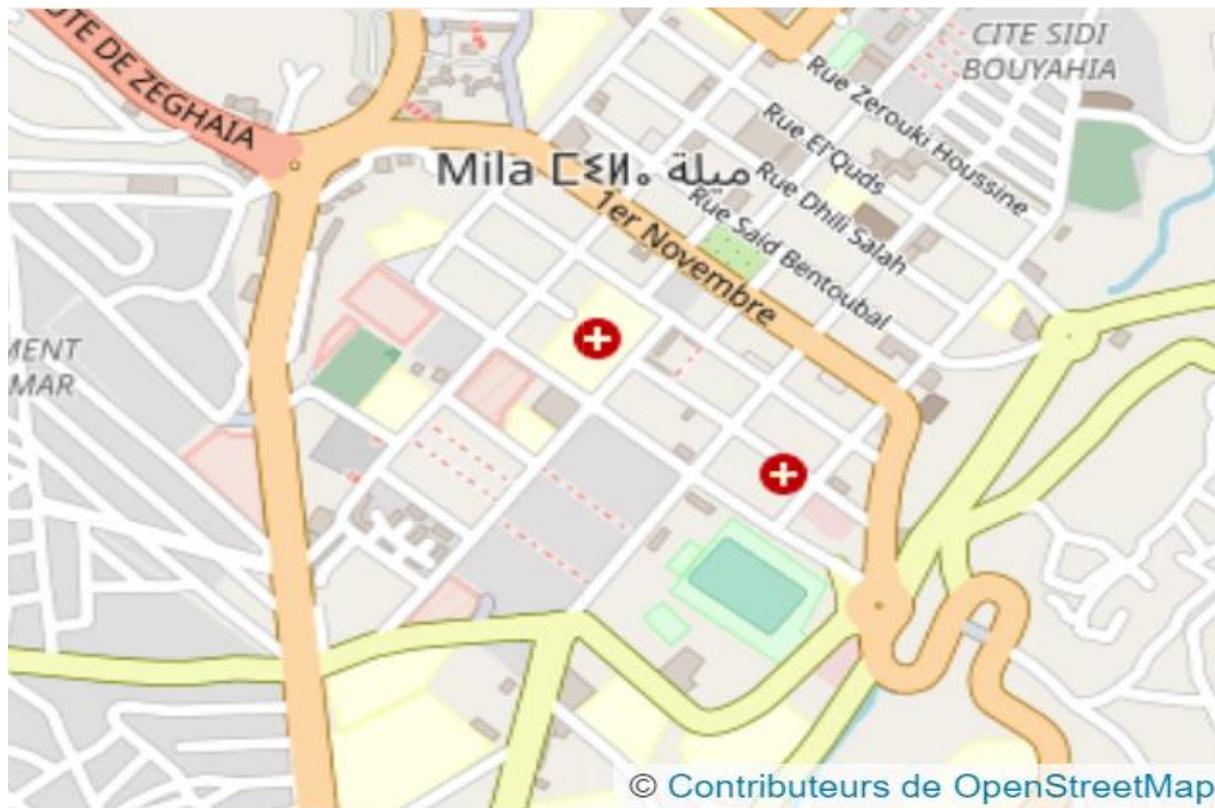


Figure 3.2: Vue de la ville de Mila en OpenStreetMap.



Figure 3.3: Réseau importé dans SUMO.

### 3.2 Outils de SUMO

SUMO est un logiciel complet qui comprend plusieurs applications nécessaires à la préparation de la simulation :

- **SUMO** : La simulation microscopique sans visualisation ; application en ligne de commande.
- **SUMO-GUI** : La simulation microscopique avec une interface utilisateur graphique.
- **NETCONVERT** : Possède la capacité d'importer et générer des cartographies adaptées pour SUMO à partir de différents formats.
- **NETGENERATE** : Permet de générer en ligne de commande un réseau de plusieurs formes : en grille, en toile ou aléatoire.

De plus, SUMO a besoin de connaître les routes possibles du réseau afin de générer les mouvements des différents véhicules. Le moyen le plus simple d'en obtenir est de les éditer manuellement, mais uniquement si leur nombre n'est pas trop élevé. La démarche manuelle va donc être de créer soi-même un fichier du type **projet.rou.xml** qui va se charger de définir les types de véhicules existant ainsi que les routes possibles. Les véhicules en circulation sur la simulation vont être alors indiqués en utilisant ces deux paramètres, et de deux manières : soit sous forme de flots arrivant de manière continue pendant une certaine période, soit individuellement. Sur de larges réseaux, SUMO fournit plusieurs applications permettant de générer ce fichier en fonction de multiples critères :

- **DUAROUTER** : Cette application obtient les routes correspondantes, en cherchant le plus court chemin entre le point de départ et d'arrivée suivant l'algorithme de Dijkstra.
- **DFROUTER** : S'appuie sur les mesures effectuées par les détecteurs.
- **DJTRROUTER** : Reprend des principes similaires, mais utilise des probabilités de tourner afin de générer les routes.
- **ACTIVITYGEN** : Se repose sur une description de la population vivant sur le réseau : nombre d'habitants, de ménages, la probabilité pour un adulte d'avoir une voiture, d'utiliser les transports en commun, description des zones d'activité de la ville (ex : école), etc. [13]

### 3.3 Lancement d'une simulation SUMO

SUMO met à disposition deux applications pour exécuter une simulation. Notons que ces deux méthodes offrent les mêmes résultats statistiques demandées en fin de la simulation:

- Au moyen de SUMO où la dynamique des véhicules sur le réseau routier est invisible. En effet, la simulation est opérée par des calculs en arrière-plan et n'affiche que quelques notifications numériques sur écran d'interaction.
- Via SUMO-GUI, qui permet de voir les véhicules en mouvement sur le réseau routier à travers l'interface graphique.

Avant lancement d'une simulation, il est nécessaire de préciser les éléments du scénario. SUMO impose deux informations de base pour toute simulation, à savoir : La topologie du réseau et le modèle de trafic.

La topologie du réseau est une configuration sur un fichier communément nommé `nom-perso.net.xml`. Quant au modèle du trafic, il est également défini sur des fichiers xml. Ces fichiers présentent des descriptions de flux de véhicules, des tronçons à parcourir pour créer des chemins. Ces fichiers de routage des véhicules sont communément nommés `nom-perso.Rou.xml`.

Il est aussi possible de rajouter un fichier ou un ensemble des fichiers supplémentaires décrivant des fonctions de simulation optionnelles tel que des capteurs ou une imagerie de fond de simulation aérienne. Ces fichiers sont communément nommés `nom-perso.add.xml`. D'autres informations permettent de créer des véhicules de différentes caractéristiques et même de décrire le type de conducteur.

Pour lancer la simulation, ces fichiers sont généralement listés sur un autre fichier xml dit de configuration de la simulation, communément nommé `nom-perso.sumocfg.xml` [2]. C'est le fichier regroupant deux fichiers Route file et Net file, Sumocfg est donc le fichier de configuration de SUMO [14]

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <configuration>
3   <input>
4     <net-file value="milanewmap.net.xml"/>
5     <route-files value="milanewmap.rou.xml"/>
6     <additional-files value="mila.poly.xml"/>
7   </input>
8   <time>
9     <begin value="0"/>
10    <end value="500"/>
11  </time>
12  <output>
13    <netstate-dump value="Milanew.xml"/>
14    <fcd-output value="Milanew.xml"/>
15    <summary-output value="Milanew.xml"/>
16  </output>
17  <Random>
18    <random value="true"/>
19  </Random>
20  <processing>
21    <ignore-route-errors value="false"/>
22    <time-to-teleport value="-1"/>
23    <ignore-accidents value="false"/>
24    <collision.action value="none"/>
25    <collision.check-junctions value="true"/>
26  </processing>
27  <report>
28    <verbose value="false"/>
29    <duration-log.statistics value="true"/>
30    <duration-log.disable value="false"/>
31    <no-step-log value="true"/>
32    <xml-validation value="never"/>
33  </report>
34 </configuration>
35
```

Figure 3.4: Exemple de « Sumocfg file ».

### 3.4 Construction de Réseau

Il existe différentes façons de créer ou importer des réseaux de transport dans SUMO.

#### 3.4.1 Création à partir des éléments de l'infrastructure

Un réseau SUMO est constitué de liens ("edge"), une ou plusieurs voies ("lane") par lien, de carrefours ("junction") et de connections ("connection") entre liens .

Ensuite l'application NETCONVERT est nécessairement sollicitée pour compiler et combiner ces descriptions et fournir la description correspondante du réseau sur le fichier.net.xml. Ces éléments et leur format sont décrits dans des fichier .xml illustré ci-dessous.

- **Node File:** C'est un fichier composé d'un ensemble de nœud du réseau routier, ces derniers sont définis par les coordonnées x et y et un identificateur (id)[14]

```
<node id="1076433287" x="1188.28" y="2081.96" type="priority"/>
<node id="1076433299" x="1090.48" y="2159.70" type="priority"/>
<node id="1076434221" x="1191.12" y="2127.35" type="traffic_light" tl="1076434221"/>
<node id="1076434247" x="1195.26" y="2080.35" type="priority"/>
<node id="1076434258" x="1182.41" y="2085.30" type="priority"/>
<node id="1793149712" x="1728.84" y="1804.71" type="priority"/>
<node id="1793149721" x="1685.19" y="1751.21" type="priority"/>
<node id="1793150163" x="1637.06" y="1689.87" type="right before left"/>
```

Figure 3.5: Exemple des «nodes » .

- **Edge file** : C'est un fichier qui relie les nœuds par des routes. Il est spécifié par un identifiant unique (id), un nœud source, nœud destination, priorité, nombre de voies et la vitesse maximal autorisée [14]

```
</type>
<edge id="-241190492#1" from="461891447" to="1818375026" priority="12" type="highway.primary" numLanes="2" speed="27.78"
  <param key="ref" value="RN 79"/>
</edge>
<edge id="-241190492#2" from="1818375033" to="461891447" priority="12" type="highway.primary" numLanes="2" speed="27.78"
  <param key="ref" value="RN 79"/>
</edge>
<edge id="-241190492#3" from="461891437" to="1818375033" priority="12" type="highway.primary" numLanes="2" speed="27.78"
  <param key="ref" value="RN 79"/>
</edge>
```

Figure 3.6: Exemple des « edges » .

- **Les fichiers des intersections (Junction)** : une collection de nœuds ou de jonctions, présentant de nombreuses propriétés spécifiant sa description physique, son emplacement, ses liens aux tronçons et leurs voies. [2]

```
<junction id="1822336515" type="priority" x="999.16" y="577.69" incLanes="-171153044#0_0"
  intLanes=":1822336515_0_0" shape="999.16,577.69 996.75,575.58 999.16,577.69">
  <request index="0" response="0" foes="0" cont="0"/>
</junction>
```

Figure 3.7: Exemple de « jonction ».

- **Les fichiers pour les plans des feux de signalisation (Traffic Light Programs)** : identifient les feux du réseau et leur plans [2].

```
<tlLogic id="1818375033" type="static" programID="0" offset="0">
  <phase duration="39" state="rrrrGGGgrrrrGGGg"/>
  <phase duration="10" state="rrrryyyyrrrryyyy"/>
  <phase duration="39" state="GGgrrrrGGgrrrr"/>
  <phase duration="10" state="yyyyrrrryyyyrrrr"/>
</tlLogic>
```

Figure 3.8: Exemple de programme « tls ».

- **Les fichiers de liaison (connection)** : précise l'ensemble de connexions possibles entre les tronçons et les voies sur route. Ils décrivent comment, à partir de voies spécifiques entrant à un nœud ou une jonction, il est possible d'en sortir [2].

```
<connections version="1.6" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://sumo.dlr.de/xsd/connections_file.xsd">
  <connection from="-167997207#0" to="241372445#5" fromLane="0" toLane="0"/>
  <connection from="-167997207#0" to="241190591#0" fromLane="0" toLane="0"/>
  <connection from="-167997207#0" to="-241372445#4" fromLane="0" toLane="0"/>
  <connection from="-167997207#0" to="167997207#0" fromLane="0" toLane="0"/>
```

Figure 3.9: Exemple de programme de connexion.

- **Net file** : C'est un fichier généré à partir de la commande NETCONVERT qui prend en argument les fichiers edg.xml et nod.xml. La commande de NETCONVERT permet de lire ces fichiers et de les convertir dans le format SUMO. [4]



pour gérer leur déplacement à travers le réseau (i.e : la rue d'entrée, la rue de sortie et toutes les routes par lesquelles ils vont passer pour atteindre la destination finale. Cette description est réalisée sur un fichier .rou.xml comme illustré sur la figure suivante

```
<vehicle id="3" depart="3.00">
  <route edges="438862994#5 438862994#6 438862994#7 438862994#8 438862994#9 309406324 309406322 92826349#5 92826349#6 92826349#7 92826349#8 170686311#0 -171152947"/>
</vehicle>
<vehicle id="5" depart="5.00">
  <route edges="-386236597 386236619#1 386236619#2 -386236609#0 167999564#9 167999564#10 170687249#4 170687249#5 170687249#6 170687249#7 170687249#8 170687249#9 694236521 -170687262 2"/>
</vehicle>
<vehicle id="6" depart="6.00">
  <route edges="-170686296#2 -170686296#1 -170686296#0 171153026#6 402167126 170681020#1 170681020#2 170681020#3 170681019 241190492#0 241190492#1 241190492#2 241190492#3 438862994#5 438862994#6 438862994#7 438862994#8 438862994#9 309406324 309406322 92826349#5 92826349#6 92826349#7 92826349#8 170686311#0 -171152947"/>
</vehicle>
<vehicle id="7" depart="7.00">
  <route edges="237301276#3 237301276#4 -171152979#2 -171152979#1 -171152979#0 167999454#1 167999454#2 309406326#0 170687253#1"/>
</vehicle>
<vehicle id="8" depart="8.00">
  <route edges="-386236609#1 -386236609#0 167999564#9 167999564#10 170687249#4 170687249#5 170687249#6 170687249#7 170687249#8 170687249#9 694236521 -170687262 2"/>
</vehicle>
<vehicle id="10" depart="10.00">
  <route edges="386231589#0 386236572#1 386236589#1 -171152991#2 -171152991#1 -171152991#0 -694230471 170687241#1 170687249#6 170687249#7 170687249#8 170687249#9 694236521 -170687262 2"/>
</vehicle>
```

Figure 3.11: Exemple de Route file.

Il existe 7 méthodes qui nous permettent de créer ce fichier:

- Définir le trafic à la main.
- Spécification du trafic aléatoire.
- Utilisation de trajets pour générer des itinéraires.
- Utilisation de points d'observation pour générer des itinéraires.
- Utilisation de sources de flux et de probabilités de virage pour générer des itinéraires.
- Utilisation de matrices Origin / Destin (OD) pour générer des itinéraires.
- Génération de la demande basée sur l'activité. [2]

### 3.5.1 Définir le trafic « à la main »

Il est possible aussi de définir le fichier de demande manuellement ou d'éditer les fichiers générés avec un éditeur de texte. Avant de commencer, il est important de savoir qu'un véhicule en SUMO est défini via 3 types d'informations :

- celles relatives au type de véhicule décrivant les propriétés physiques du véhicule.
- celles de l'itinéraire que le véhicule doit emprunter.
- et celles relatives au véhicule lui-même.

Les itinéraires et les types de véhicules peuvent être définis en communs à plusieurs véhicules. A défaut de précision, un type de véhicule par défaut est utilisé. Cependant, la précision du parcours des véhicules est impérative. Le conducteur d'un véhicule n'est pas modélisé explicitement, mais

SUMO présente quelques propriétés aux véhicules qui simulent son attitude et comportement. Pour la simulation de personnes qui se promènent ou chevauchent les véhicules, des définitions supplémentaires sont nécessaires. [2]

### 3.5.2 Spécification du trafic via outils de SUMO

SUMO présentent différentes applications pour créer les parcours :

Le moyen le plus simple de générer du trafic pour une simulation SUMO consiste à utiliser l'outil **randomTrips**. Il permet de générer un ensemble de trajets aléatoires pour un réseau donné. Il sélectionne les tronçons source et destination des déplacements, soit uniformément au hasard, soit avec une distribution au choix. Les trajets qui en résultent sont stockés dans un fichier XML qui peut être directement utilisé lors des simulations. Les parcours sont distribués uniformément dans un intervalle défini par des heures de début et de fin configurables. Leur nombre est défini par un taux configurable. Chaque parcours a un identifiant composé d'un préfixe et d'un numéro courant.

Par exemple, la commande suivante : **randomTrips.py -n nom.net.xml -e 5000 -r nom.rou.xml** génère un fichier d'itinéraire pour le réseau de topologie indiqué par **nom.net.xml**, avec des choix de parcours sur le fichier **nom.rou.xml**, commençant à la seconde 0 jusqu'à 5000 secondes. Il existe de nombreux paramètres qui peuvent être ajustés afin de configurer différentes distributions de probabilités et d'autres fonctionnalités. DUAROUTER est un autre outil responsable de l'importation des itinéraires ou de leurs définitions à partir d'autres packages de simulation et du calcul des itinéraires. Une autre possibilité est JTRROUTER, qui peut être utilisée si on veut modéliser le trafic statistiquement, en utilisant des flux et des pourcentages de rotation aux jonctions. Une troisième application, OD2TRIPS aide à convertir les matrices OD (matrices origine / destination) en trajets. Finalement, DFROUTER, calcule les itinéraires à partir des mesures prises à partir des points d'observation sélectionnés. Le programme ACTIVITYGEN peut être utilisé pour transformer les statistiques démographiques en fichier de trajet, puis utiliser

DUAROUTER pour générer une demande de trafic (i.e ne génère pas lui-même le fichier de route mais peut être utilisé conjointement avec DUAROUTER). [2]

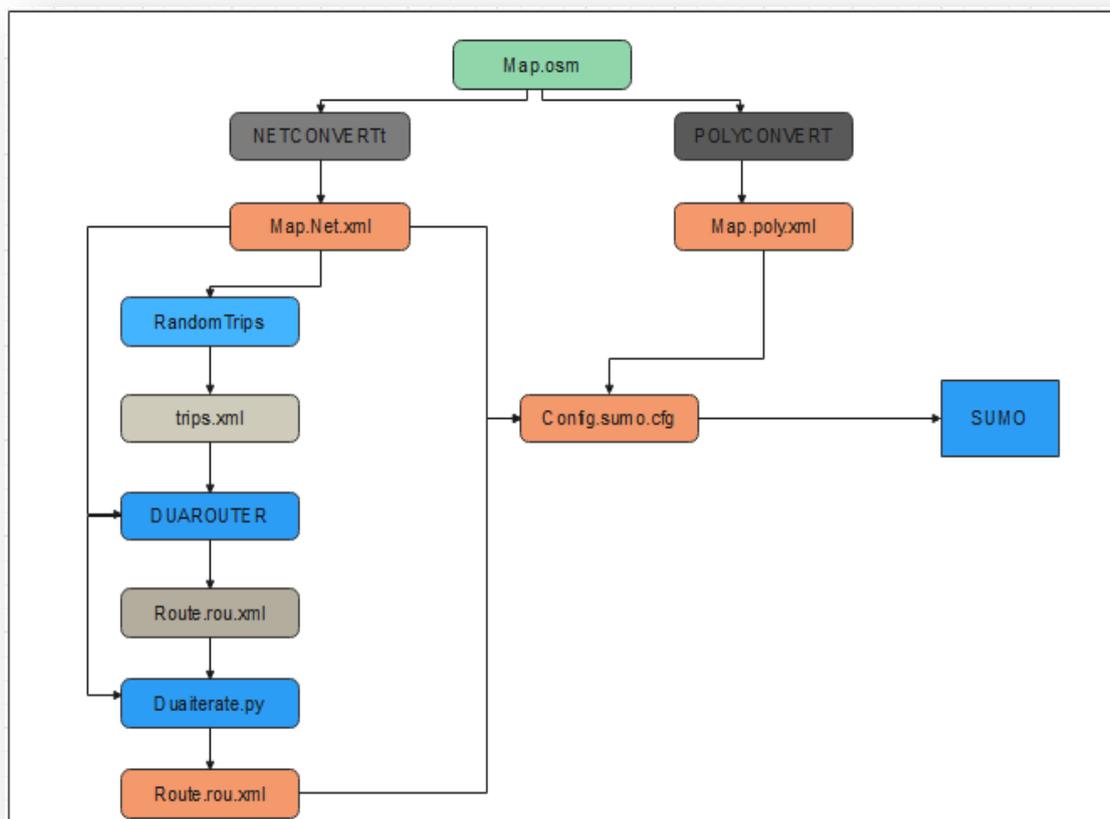


Figure 3.12:Fichiers de configuration de la circulation urbaine.

### 3.6 Les résultats des simulations

Avec l'utilisation des fichiers net.xml et rou.xml uniquement, la simulation peut être exécutée. De plus, et à l'aide de SUMO-GUI l'utilisateur peut voir les véhicules circuler dans le réseau et zoomer sur certaines parties de la structure pour obtenir une vue plus détaillée du flux de véhicules.

Mais ce ne sont pas les seuls résultats obtenus car SUMO offre une multitude de traces des simulations caractérisés par :

- des fichiers .xml volumineux car détaillent les résultats à la seconde ou au véhicule près.
- couvrent les résultats des simulations depuis leur début jusqu'à la fin.

- les résultats sont à la demande : soit spécifiques à un sujet particulier (par véhicule, par route, . . .), soit génériques sorte de statistique de synthèse.

Pour obtenir ces fichiers trace, il suffit de préciser les sorties souhaitées via les options fournies par SUMO et ce sur le fichier `.sumocfg.xml` (en plus du `.net.xml` et de `.rou.xml`).

### 3.7 Contrôle des simulations via TRACI

SUMO offre également une interface nommée "**TraCI(Traffic Control Interface)**" qui permet de connecter le simulateur avec d'autres applications externes pour la supervision, la récupération des résultats et la modification de simulation en cours d'exécution.

#### 3.7.1 Utiliser TraCI :

TraCI utilise une architecture client / serveur basée sur TCP pour fournir un accès à SUMO. Ainsi, SUMO agit en tant que serveur démarré avec des options de ligne de commande supplémentaires : `--remote-port <INT>` ou `<INT>` est le port que SUMO écoute pour les connexions entrantes. Lorsqu'il est lancé avec l'option `--remote-port <INT>`, SUMO prépare uniquement la simulation et attend que toutes les applications externes se connectent et prennent le contrôle. Veuillez noter que l'option `--end <TIME>` est ignorée lorsque SUMO s'exécute en tant que serveur TraCI. SUMO s'exécute jusqu'à ce que le client demande la fin de la simulation. Lors de l'utilisation de SUMO-GUI en tant que serveur, la simulation doit être lancée à l'aide du bouton de lecture ou en définissant l'option `--start` avant que les commandes TraCI ne soient traitées. [14]

#### 3.7.2 Plusieurs clients :

Le nombre de clients pouvant se connecter peut-être indiqué en option supplémentaire `--num-clients <INT>`, où 1 correspond à la valeur par défaut. Notant que dans les scénarios multi-clients, vous devez spécifier explicitement l'ordre d'exécution des clients à l'aide de la commande `SetOrder`. Chaque client doit spécifier une valeur entière unique (mais arbitraire) et les commandes du client seront traitées dans l'ordre, de la valeur la plus basse à la valeur la plus élevée dans chaque étape de la simulation. Les clients sont automatiquement synchronisés après chaque étape de la simulation. Cela signifie que la simulation n'avance pas à l'étape suivante tant que tous les clients n'ont pas appelé la commande '**simulationStep**'. De plus, la commande **simulationStep** ne restitue le contrôle au client qu'une fois la simulation avancée. [14]

### **4. Conclusion**

SUMO a fourni jusqu'à présent la simulation parfaite grâce aux options et aux applications qu'il comprend. Exploité par des ingénieurs de transport ou en informatique, il a facilité de nombreuses recherches de solutions aux problèmes du réseau routier.

# Chapitre 04 : Contrôle des feux tricolores via PSO à trois essais coopératifs sur SUMO

---

## 1. Vue générale du problème traité

Divers problèmes causés par la circulation automobile comme la pollution, l'embouteillage, la sécurité, le stationnement...etc sont présents tous les jours dans la plupart des villes du monde. Pour réduire l'impact de ces problèmes, les chercheurs s'accordent souvent à dire qu'une bonne programmation des feux de signalisation améliore la circulation des véhicules à travers les villes [17]. Depuis des décennies déjà, les efforts des chercheurs en ingénierie des transports se sont concentrés sur la modélisation microscopique précise des flux de trafic et la programmation de cycles de feux de circulation adaptés. Par conséquent, l'utilisation de méthodes intelligentes s'est avérée utile pour l'optimisation de la programmation des cycles de feux de circulation.

En Algérie, la politique de transport a muni les infrastructures routières de feux de signalisation, cependant la stratégie de contrôle employé est très obsolète. Elle repose généralement sur un partage équitable des phases du cycle d'un feu et l'ajustement via onde verte aux meilleurs cas. L'infrastructure routière de la wilaya de Mila ne fait pas exception à cette situation.

L'objectif de notre travail est de proposer une solution au problème de régulation du trafic via feux de signalisation. Nous déployons cette solution sur le réseau routier du centre-ville de MILA sur les feux tricolores en plusieurs point d'intersection de l'infrastructure routière. Cette proposition est basée sur l'algorithme d'optimisation par essaim particulière PSO via son extension en trois essais particuliers coopératifs. Elle est mise en œuvre et testée sur le simulateur SUMO. Les scénarios sont élaborés, en termes de quantité du flux lors des déplacements, en utilisant des statistiques provenant de la direction des transports de la wilaya de Mila.

Le présent chapitre explique la démarche suivie pour traiter le problème et détaille notre proposition.

### 2. Formalisation du problème

Commençant par la fonction utilité, notre travail vise à minimiser les durées moyennes de déplacement DD des véhicules en ville (sur les zones traitées). Les durées de parcours sont mesurées pour chaque véhicule depuis le démarrage d'un point de départ jusqu'à atteinte de la destination. Formellement, elles sont le résultat de la somme des durées de circulation en vitesse libre DVL et des durées de circulation en vitesse moindre que celle désirée DVM. Ces DVMs incluent les temps d'attente à l'arrêt (à vitesse  $V=0$ ) devant les feux et les durées de circulation dans des embouteillages ou des pletons. De plus, nous voulons réduire les files devant les feux. Elles peuvent être mesurées en termes de nombre de véhicules, de distance qu'elles occupent ou encore de temps d'arrêts TA qu'elles engendrent lors des attentes.

Pour ce faire, nous agissons au niveau de quelques feux de circulation de la ville de Mila. Les programmes des feux en termes de succession des autorisations de passages ne sont pas modifiés puisqu'ils sont conçus par des ingénieurs de transport afin d'assurer la sécurité routière en respect de la matrice de conflit sur chaque carrefours. Néanmoins, nous pouvons modifier les phases de chaque feu indépendamment.

Il s'agit alors de déterminer intelligemment les durées de phases associées à chaque feu parmi le nombre extraordinaire de durées possibles afin de minimiser les temps d'attente devant les feux et les durées de déplacement des véhicules sur la zone traitée.

Cependant, on est contraint de respecter les temps minimum et maximum des phases. En effet, en ingénierie des transports, il est indiqué que les phases ne doivent pas être trop longues ce qui poussent les conducteurs enfreindre les droits de passage et à prendre ainsi des risques de causer des accidents. Elles ne doivent pas être trop courtes non plus pour réduire les nombres d'arrêts /redémarrage qui mènent à une circulation non-confortable et à risques de collisions. De même, l'ordre des phases selon le programme de chaque feu de signalisation est impératif.

Quant aux variables qui modélisent le problème, il s'agit principalement des déplacements des véhicules en ville en termes de choix des itinéraires, du nombre de véhicules sur l'infrastructure, des conditions de déplacement, des types de véhicules, des moments d'arrivées devant les feux, ...etc et elles sont établies et traitées via simulation. Il est important qu'elle puisse reproduire l'interaction entre ces éléments selon les conditions réelles. Les résultats en termes des temps de déplacement et temps d'attente en sont récupérés pour traitement via la méthode d'optimisation choisie.

Notons que les feux de signalisation déployés sur la wilaya de Mila sont programmés au préalable pour suivre un cycle composée de phases inchangée. Ils ne permettent pas un contrôle adaptatif en temps réel. C'est pourquoi, la solution est calculée en temps différé et mise en œuvre pour des flux de circulation de même densité.

La solution  $X^*$  détermine les durées des phases du cycle de chaque feu sur les intersections considérées.

$$\min \left( \sum_i^N DD_i(\Phi) + \sum_i^N TA_i(\Phi) \right) / N$$

Sous contraintes

$$\varphi_f \in \Phi$$

$$\varphi_{min} < \varphi_f < \varphi_{max}$$

$$\sum_{h=1}^H \varphi_f = C_f = cst$$

$$\max C_f \ll T$$

$i$  : indice d'un véhicule sur la zone étudiée

$f$  : indice des feux contrôlés

$T$  : durée de l'étude

$N$  : nombre de véhicules en déplacement durant  $T$

$DD$  : Durée de parcours mesurée pour chaque véhicule  $i$

$TA$  : Temps d'attente mesurée pour chaque véhicule  $i$

$\varphi_f$  : une phase du programme du feu de signalisation

$\Phi$  : ensemble des phases concernés par le contrôle

$\varphi_{min}$  et  $\varphi_{max}$  : durée minimale et maximale respectivement d'une phase

$H$  : nombre de phases d'un même feu

$C_f$  : durée de cycle du feu de signalisation  $f$

### 3. Stratégie d'optimisation

La stratégie d'optimisation se compose essentiellement de deux parties principales en interaction :

1) l'algorithme d'optimisation

2) une procédure de simulation.

La partie optimisation est réalisée au moyen de l'algorithme d'optimisation par trois essais particuliers coopératifs, qui a été spécialement adapté pour trouver un cycle optimal des programmes pour les feux de circulation. Elle fait appel, lors de l'une de ces étapes, au simulateur choisie, i.e SUMO pour évaluer chaque particule  $i$  traitée. SUMO remet les traces résultant de l'exécution du scénario au TSC-PSO. La figure suivante illustre cette stratégie.

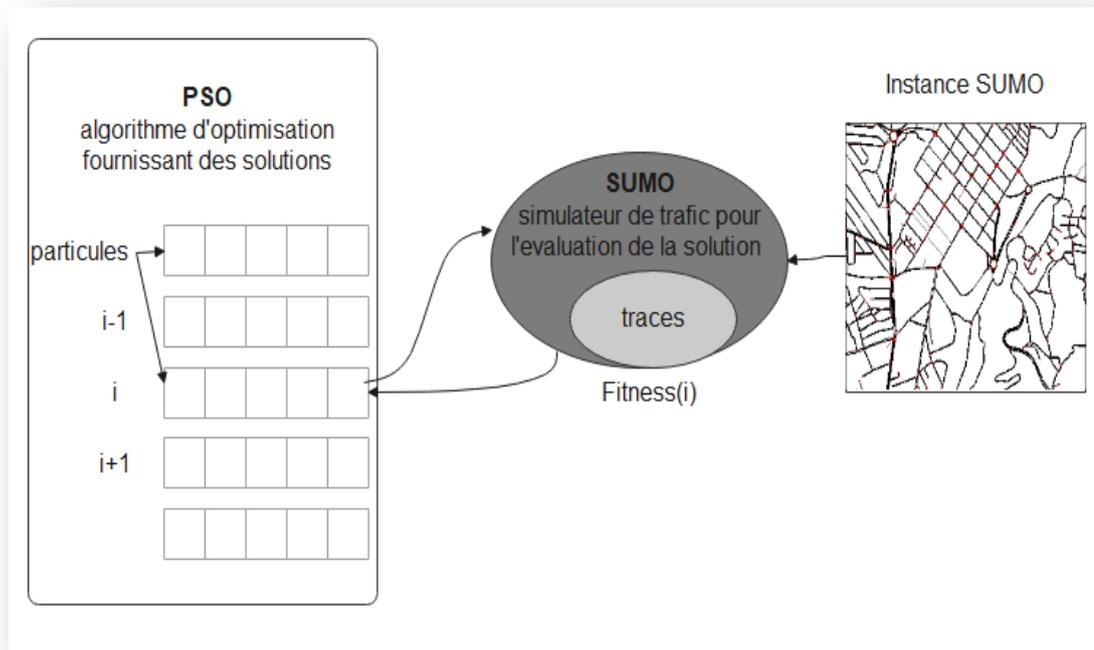


Figure 4.1: Stratégie d'optimisation.

### 3.1 Résolution via PSO à 3 essais coopératifs

#### 3.1.1 Avantage du PSO

L'utilisation de PSO au lieu d'autres techniques d'optimisation est principalement due à ce qui suit :

Premièrement, le PSO est un algorithme bien connu, démontré pour effectuer une convergence rapide vers des solutions adaptées. Il s'agit d'une propriété hautement souhaitable pour un programme de cycle de feux de circulation optimal, où de nouvelles durées des feux de circulation pourraient être nécessaires pour gérer les événements dans les scénarios de trafic.

Deuxièmement, le PSO certes heuristique mais nécessite peu de paramètres de configuration [17].

Troisièmement, les auteurs de l'extension proposée TSC-PSO [17] ont démontré qu'elle peut traiter efficacement l'optimisation via synchronisation du signal, soulager la débordement du trafic et améliorer l'efficacité du trafic urbain.

### 3.1.2 Description des particules

Etant donné que le contrôle à effectuer a pour objectif de chercher le meilleur phasage de tous les feux de la zone étudié, PSO doit calculer les phases des programmes des feux tricolores et les tester. Les particules modélisent ces différentes phases. En d'autres termes, les particules codifient les phases comme vecteurs fournis au PSO pour optimisation.

La structuration globale des feux tricolores a été codée au moyen d'un vecteur d'entiers, où chaque composante du vecteur représente une durée de phase du feu de circulation pour une intersection donnée. De cette façon, comme le montre la figure 4.2, toutes les durées de phase des feux sont successivement placées dans le vecteur de solution, mappant ainsi l'ensemble des feux de circulation traités dans un simple tableau d'entiers.

Le programme de chaque feu est cyclique, c.à.d. l'ordre d'apparition des couleurs est inchangé ainsi que les durées accordées.

La raison pour travailler avec cette représentation est double. Premièrement le simulateur SUMO lui-même fonctionne avec des valeurs entières pour représenter la séquence discrète de pas de temps (secondes) qui composent la procédure de simulation complète. Deuxièmement, de vrais feux de circulation emploient également des valeurs entières pour spécifier la durée des phases dans leurs programmes internes.

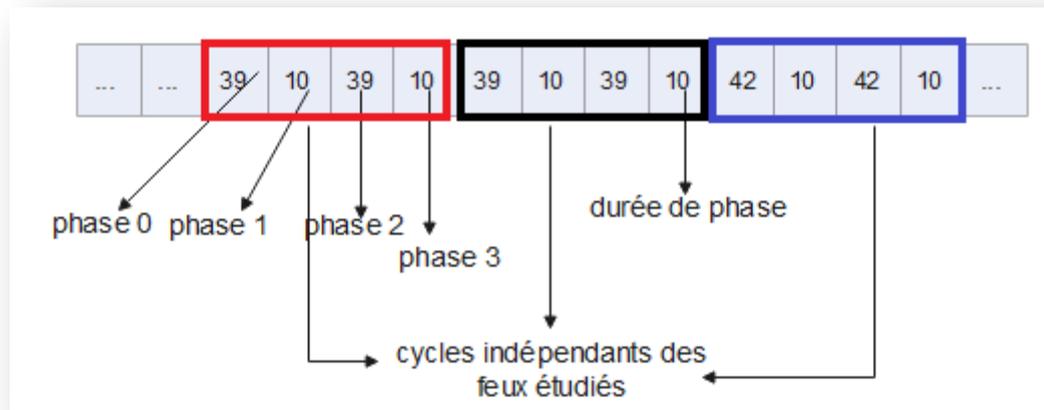


Figure 4.2: Illustration d'une particule selon le problème étudié.

Concernant la dimension de chaque particule, on a travaillé sur 6 feux de circulation. Un seul feu présente 3 phases alors que cinq autres utilisent 4 phases, ayant au totale 23 phases, d'où la dimension  $N_{dim}=23$  pour chaque particule.

### 3.1.3 Paramètres du PSO

Dans ce qui suit, on présente le choix des valeurs de quelques paramètres du PSO

- **Taille des essaims** : représente le nombre des particules de chaque essaim. Dans notre travail on a 3 essaim (deux essaims excellents, et un mauvais), chaque essaim comporte 3 particules (nombre des particules  $M_{particules} = 9$ ).
- **Initialisation des particules** : Chaque particule est initialisée aléatoirement au niveau des composantes du vecteur, sur l'intervalle [phase min, phase max].
- **Coefficients d'accélération** : Les coefficients  $c_1$ ,  $c_2$ ,  $c_3$  et  $c_4$ , permettent de pondérer les tendances des particules à suivre leur propre expérience ou celle de leurs voisins au niveau local ou global. Le choix de leurs valeurs est crucial pour le comportement du PSO. La documentation relative au PSO ne donne pas de méthode exacte pour le choix de  $c_1$  et  $c_2$  mais recommande des valeurs différentes et proches de 2.0. nous avons effectué la même procédure empirique que les auteurs de [2] pour choisir leur valeur. En effet, L'intervalle que nous avons choisi aussi bien pour  $c_1$  et  $c_2$  est [1.6 ,2.3]. Nous avons conduit quelques essais en simulation pour des valeurs prises aléatoirement dans cet intervalle.

### 3.1.4 Étapes du TSC-PSO pour le contrôle proposé

- Les particules sont distribuées sur 3 essais (2 essais excellents, et l'autre mauvais)
- Chaque essaim est initialement composé d'un certain nombre de particules ou solutions potentielles de phase aléatoires. Ces valeurs sont comprises dans l'intervalle de temps [10, 60] constituent la plage de durées possibles (en secondes) qu'un feu de circulation peut être maintenu sur une couleur de signal.

Comme pour la version standard, l'algorithme opère plusieurs itérations, que nous avons fixées à 10 dans lesquelles il répète les étapes suivantes :

- la fonction fitness est évaluée pour chaque particule en utilisant SUMO. Chaque vecteur de solution ou cycle des programmes de feux de circulation, est évalué en tenant compte des informations obtenu à partir des événements survenus pendant la simulation. La fonction objectif est la somme du temps d'attente moyen et la durée moyenne de déplacement des véhicules. Voir la section suivante pour plus de détails sur l'évaluation de cette fonction.
- comparer la nouvelle valeur de fonction avec l'ancienne valeur pour chaque particule:
  - ✓ S'il y'a une modification : mettre à jour la valeur de la fonction et garder trace de la particule actuelle.
  - ✓ Sinon, les valeurs ne change pas.
- Chercher et extraire la meilleure position locale, i.e la meilleure position parmi le voisinage locale de chaque essaim.
- Chercher et extraire la meilleure position globale, i.e la meilleure position entre tous les essais.
- Les déplacements des particules  $v_i$  et leurs nouvelles positions  $x_i$  sont calculés et mises à jour à chaque itération selon les équations (1.8) et (1.9) (1.10) (1.11) vues au chapitre 1. Elles sont mises à jour en respect des contraintes imposées par la conception des feux tricolores. En effet, si les calculs donnent des résultats qui renvoient les particules hors des solutions potentielles alors PSO opère une troncature pour ramener les valeurs à l'intérieur du domaine de recherche spécifié dans les contraintes.

- De plus, en fin de chaque itération, le PSO coopérative peut opérer une permutation entre essais et c'est ce qui fait sa force à chercher sur tous le domaine des solutions faisables. L'échange se fait entre toutes les particules d'un essaim excellent et celles du mauvais essaim. Evidemment, même le résultat de l'historique des meilleures particules et leurs évaluations sont permutées. Cet échange est opéré sous condition que la solution potentielle de l'itération en cours n'améliore pas l'ancienne solution retenue au préalable.

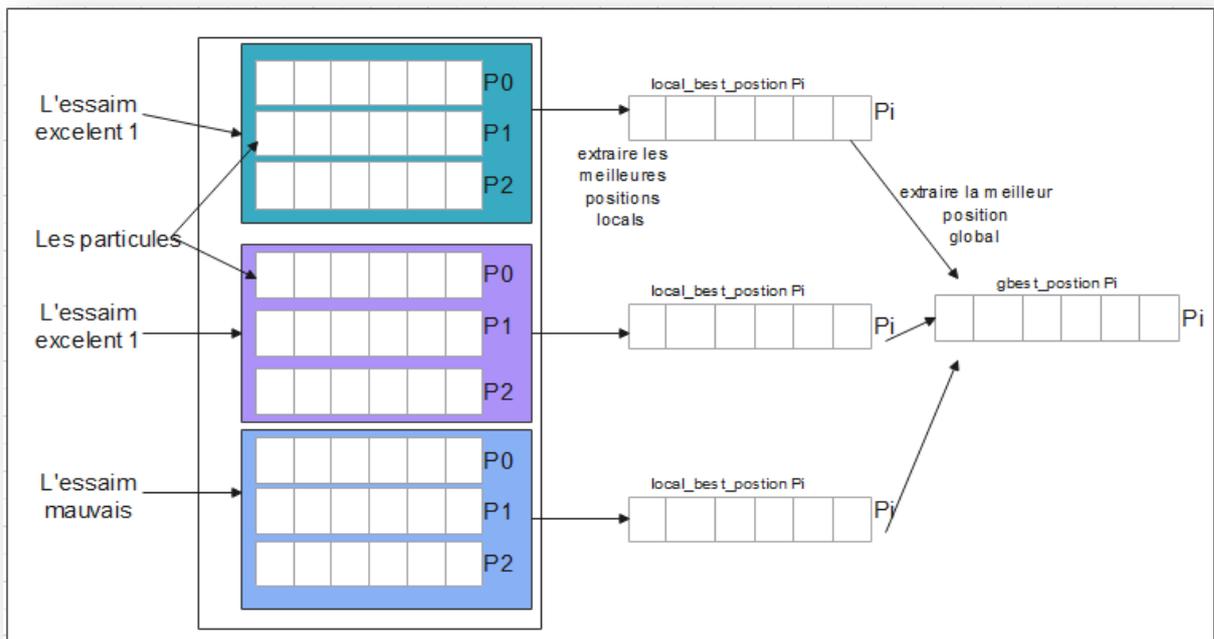


Figure 4.3: Stratégie d'optimisation avec 3 essais particulaire.

#### 4. Mise en œuvre de SUMO dans la démarche de contrôle

La tâche de SUMO est d'évaluer chaque programme du cycle

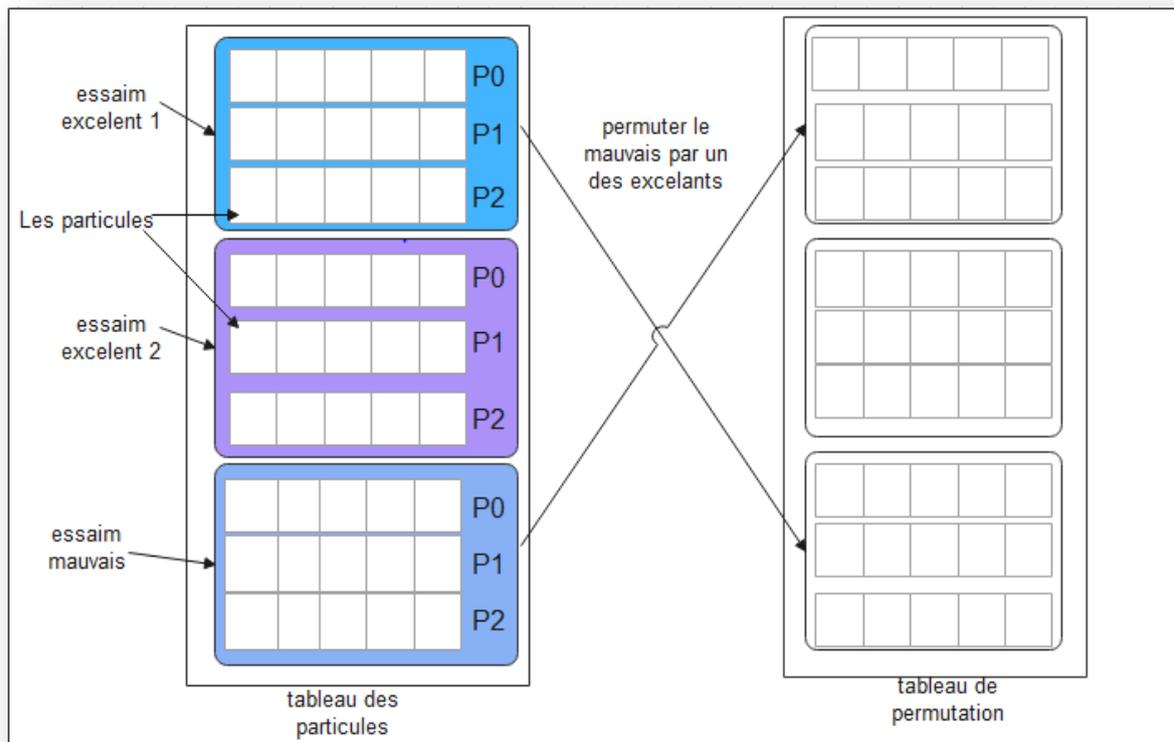


Figure 4: L'étape de permutation entre un mauvais et un excellent essaim.

Des feux tricolores, proposés par TSC-PSO durant l'évolution des calculs. Sumo permet de modéliser le trafic réel par des instances de scénarios et de les contrôler par des phases de feux calculées sur les particules de PSO.

Ainsi, lorsque PSO génère une nouvelle solution, elle est immédiatement utilisée pour mettre à jour le programme du cycle : i.e SUMO est lancé pour simuler l'instance de scénario dans le cadre du nouvel phasage défini par cette solution. Après la simulation, SUMO renvoie les informations globales nécessaires (la fonction fitness) pour traitement au TSC-PSO pour calculer les nouveaux vitesses et positions pour mettre à jour les particules.

##### 4.1 Préparation des scénarios

Le scénario en SUMO met en interaction les rues, les directions, les obstacles, les feux de circulation, les véhicules, les vitesses, les itinéraires, etc. C'est pourquoi, SUMO nécessite plusieurs fichiers d'entrée qui contiennent des informations sur le trafic et l'infrastructure routière à simuler.

Le réseau est importé à partir de cartes numériques populaires OpenStreetMap (OSM) et converti en un réseau SUMO (**fichier.net.xml**) au moyen d'une série de scripts fournis dans le package SUMO. Nous avons choisi OSM car il offre à la fois des données géographiques correctes et des informations sur les feux de circulation [17]. Mais il est nécessaire de lui indiquer le besoin en feux de signalisation sur le réseau. Nous avons importé le réseau avec tls (Traffic Lights) en précisant à Netconvert l'identificateur des intersections contrôllées via l'option **--tls.set** :

```
Netconvert --Net-files fichier.net.xml --tls.set id_junction1, id_junction2,...
```

Pour définir le flux de trafic sur l'infrastructure routière, nous utilisons le script randomTrips qui fait appel à Duarouter qui va générer les véhicules dans les routes d'une manière dynamique comme suit

```
randomTrips.py -n fichier.net.xml -r fichier.rou.xml -o trips.xml -e 1000 -p 1
```

Pour chaque jonction associée à un feu de signalisation, le tls de SUMO doit être caractérisé par :

**Un id** : identifiant du feu de circulation. En règle générale, l'identifiant d'un feu de circulation est identique à l'identifiant de la jonction associée.

**Le type** : Le type de feu (durées de phase fixes, prolongation de phase basée sur les écarts de temps entre les véhicules (**onde verte**) ou sur la perte de temps cumulée des véhicules en file d'attente (delay\_based)).

**ProgramID** : L'identifiant du programme de feux de circulation. En effet, un feu peut actionner l'un des programmes selon certaines conditions déterminées au préalable.

**Offset** : Le décalage horaire initial du programme. Ceci n'est faisable que pour les feu de type prolongation de phase.

Le programme de chaque feu ou **tl\_Logic** contient un cycle, et chaque cycle contient plusieurs phases. Chaque phase est définie par une **durée** (duration), et **l'état** (state) représenté par un des caractères qui signifie une couleur, entre autres :

**r** : red ('feu rouge' pour un signal - les véhicules doivent s'arrêter).

**y** : yellow ('feu jaune' pour un signal - les véhicules commenceront à décélérer s'ils sont loin de la jonction, sinon ils passent).

**g** : green ('feu vert' pour un signal, pas de priorité - les véhicules peuvent passer la jonction)

**G** : green ('feu vert' pour un signal, priorité - les véhicules peuvent passer la jonction).

### 4.2 Interaction entre SUMO et TSC-PSO

La figure suivante présente une partie du réseau sur lequel on a travaillé. Elle montre les feux de chaque jonction avec les cycles et les durées de phases.

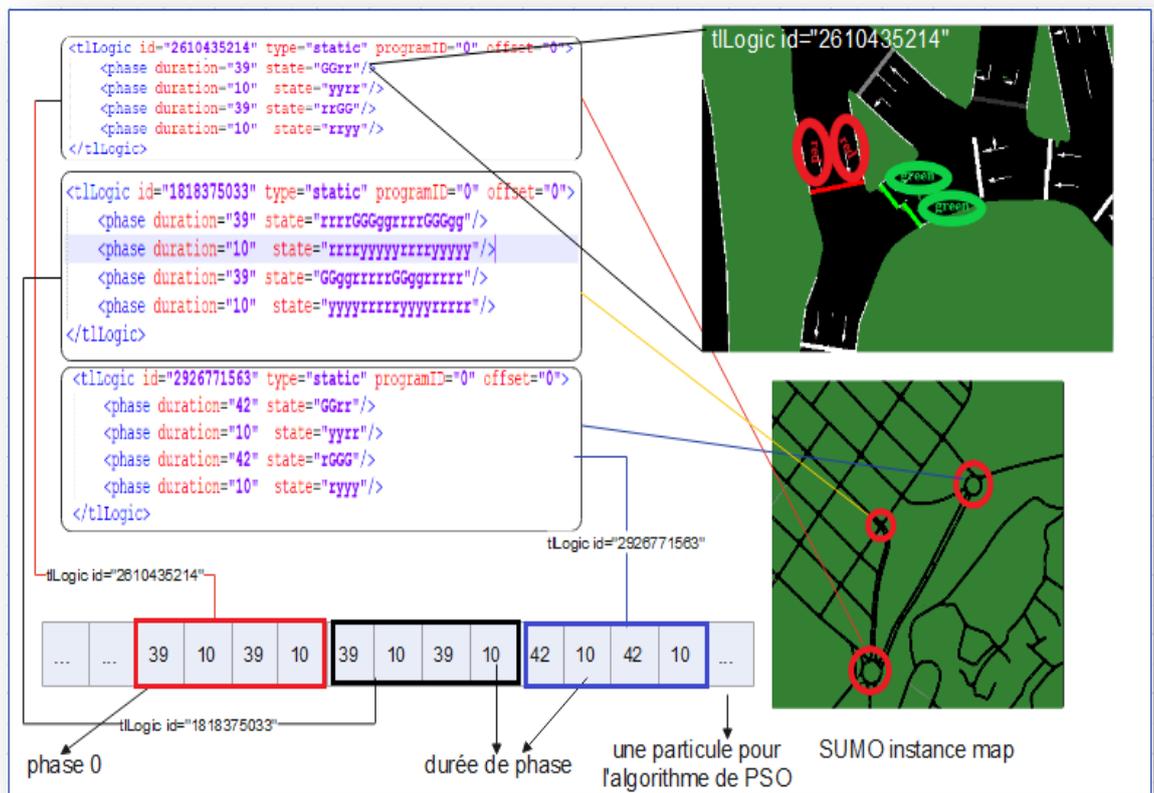


Figure 4.5: Intersection avec plusieurs feux de circulation sélectionnés à partir de la carte d'instance SUMO.

Le tl-logique avec id= 2610435214, qui correspond à une intersection de l'instance SUMO, contient quatre phases avec durées de 39, 10, 39 et 10 s. Dans ces phases, les états ont quatre couleurs, chacune correspondant à l'un des entrées de l'intersection gérées par quatre feux de circulation situés à l'intersection étant étudié.

Dans ce cas, la première phase contient l'état GGrr signifiant que deux feux de circulation sont en vert (G) et les deux autres sont en rouge (r) pendant 39 s. la phase suivante

modifie l'état des quatre voyant à yrr (y est le jaune) pendant 10 s, et ainsi de suite. La dernière phase est suivie de la première et ce cycle se répète tout au long de la simulation.

Chacun des tl-logics du scénario de SUMO exécute son propre cycle de programmation des phases au cours de la simulation indépendamment les uns des autres, constituant ainsi la mise en scène globale des feux de circulation.

Pour déterminer le meilleur plan des feux, chacune des particules doit être soumise à la simulation pour déterminer le temps d'attente qu'elle génère. Le programme de TSC-PSO initialise une simulation en chargeant le fichier de configuration du scénario qui lui-même indique les fichiers du réseau et du flux. De plus, les feux ne sont plus contrôlés par le programme des tl-logics mais par le programme de TSC-PSO. Sumo offre une interface qui permet ce contrôle en se mettant en mode serveur et attend les commandes de contrôle qui lui seront fournies via l'interface TraCI depuis le **fichier.py** sur le port (8813).

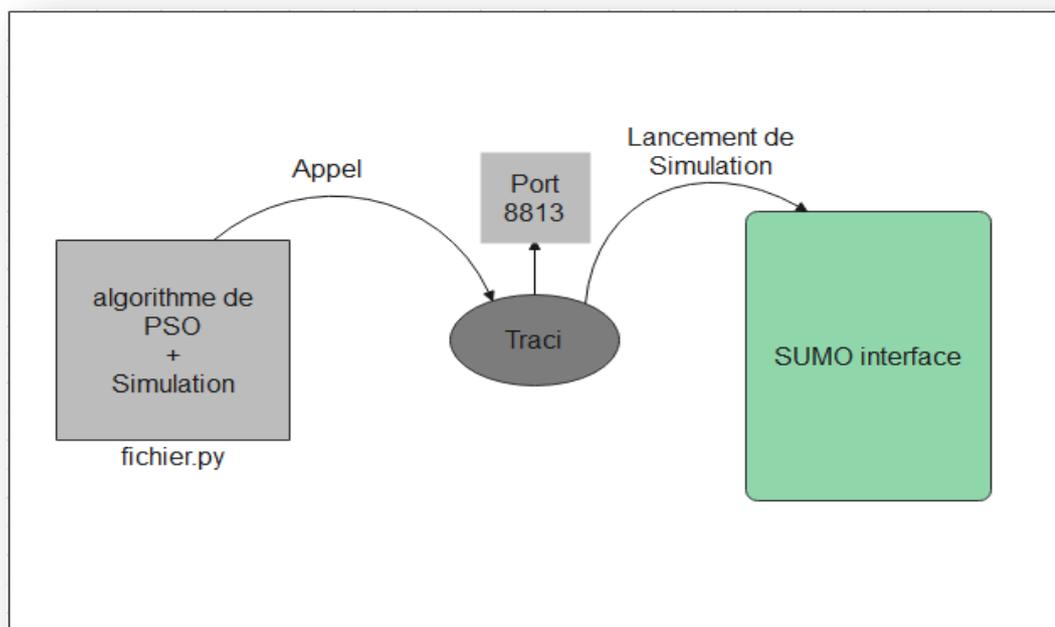


Figure 4.6: Communication entre SUMO et TSC-PSO via Traci

## 5. Organigramme de code

Les organigrammes ci-dessous présentent notre travail et la combinaison Optimisation\_Simulation :

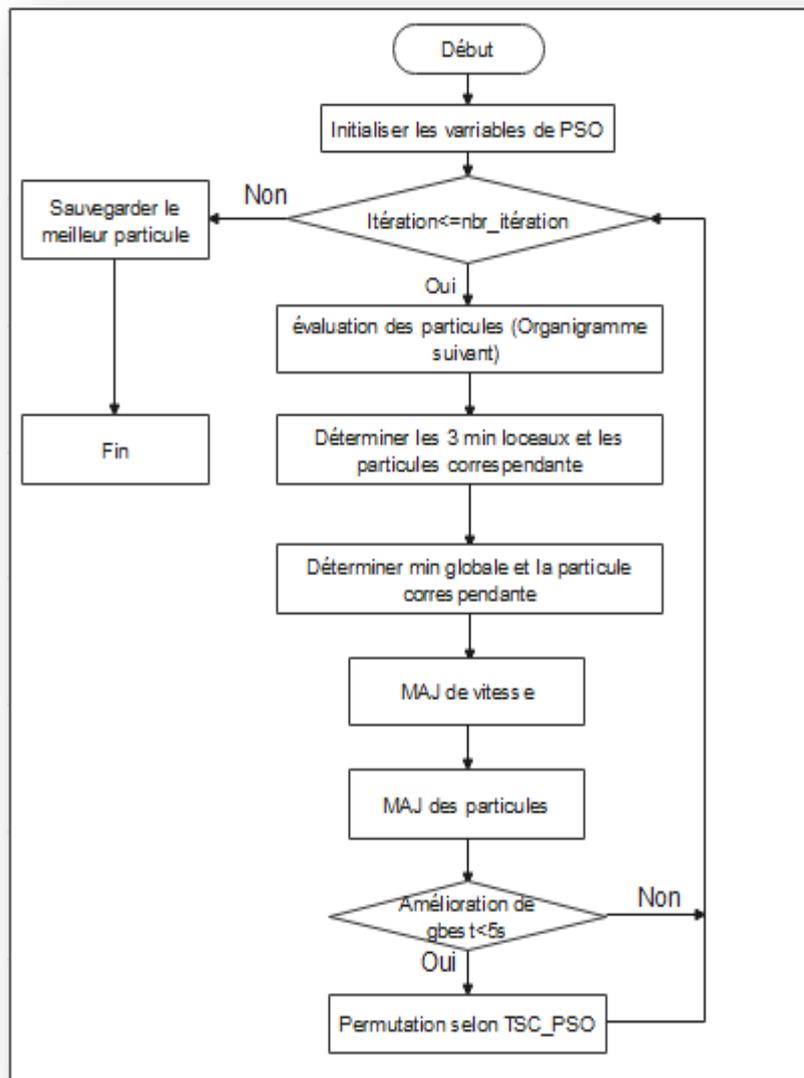


Figure 4.7: Organigramme principal.

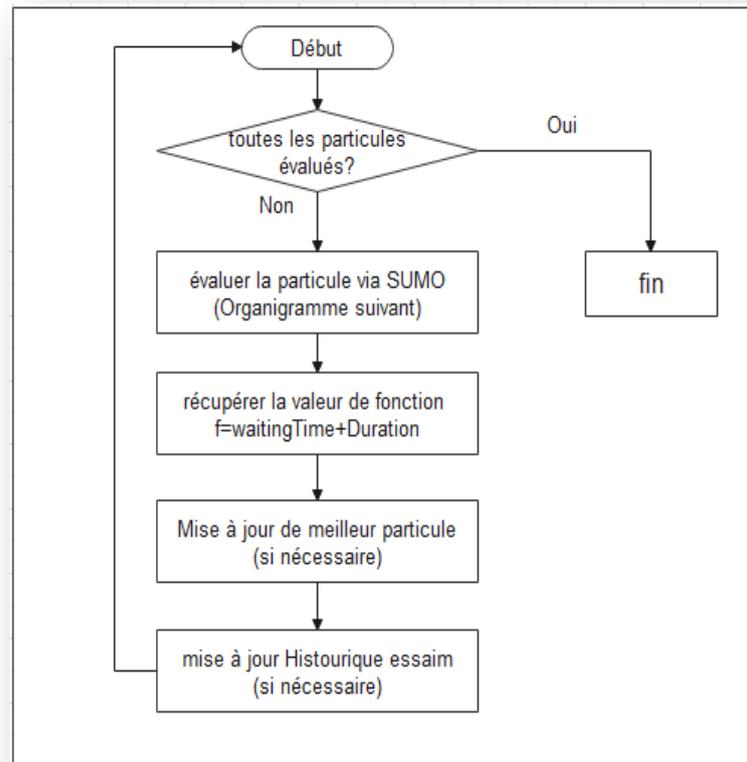


Figure 4.8: Organigramme de fonction fitness\_local.

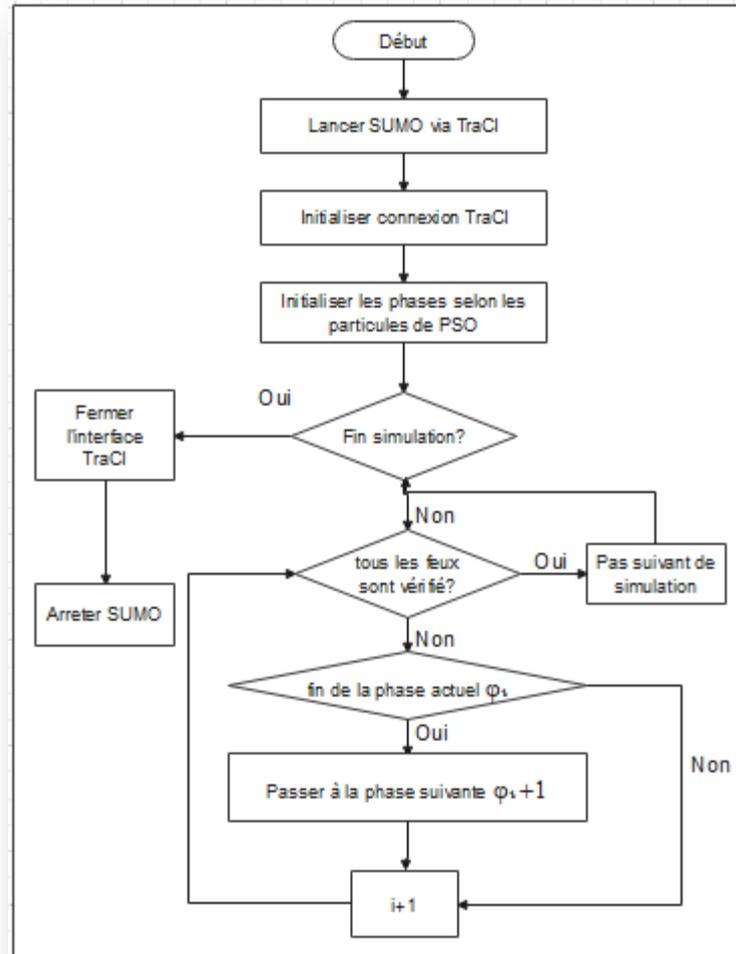


Figure 4.9: Organigramme de fonction fitness.

## 6. Statistiques sur les routes de Mila

Pour étudier et résoudre le problème de circulation dans la ville de Mila, nous avons collecté des statistiques sur le trafic routier à Mila depuis la Direction des Transports de la Wilaya de Mila faites à l'horizon 2015.



Figure 4.10: Carte du réseau de Mila

La zone des localités et communes en dehors du chef-lieu de la wilaya génère des émissions de trafic quotidien à destination la ville de Mila caractérisée par un volume journalier de déplacements de : 17 401 déplacements des différents modes de transport.

L'hypercentre de Mila affiche un volume dominant estimé à 16 127 déplacements dénombrés aux heures de pointe représentant 17.67 % du volume global de déplacements.

La première couronne du côté nord de la ville génère un volume de 11459 déplacements/jour représentant un taux de 12.56 % du volume global, ce poids est dû à la concentration de la population sur ce pôle résidentiel représenté par les cités : Nigro, Frères Boukellala , Bidi.

La région du côté ouest affiche un volume de 5092 déplacements/ jours soit respectivement 5.58 % du global emis. Cette région englobe l'extension future de la ville représentée par Ben Mahdjoub, Boulmerka et frères Ben Maamar, et l'université et la zone d'activité.

Vient en dernière position la zone de la cité administrative représentée par la cité Salah Bouttot avec tout juste un volume 3395 déplacement/jour soit un taux de 3.72 % du volume

global. Ce pourcentage de déplacements émis est significatif si on compare cette zone avec les autres zones de la ville

Les zones du côté est, représentés par les cités : Vieux Mila, sidi Esghir, frères Laabyi et 535 logements, affichent un volume de : 8 913 déplacements /jour représentant ainsi un taux de 9.77 % du volume global émis. Ce taux d'émission est la résultante du poids résidentiel de ces zones à vocation multiple, avec une part importante pour la vocation résidentielle.

La partie sud de la ville offre un volume de 7639 déplacements journaliers émis vers les différentes zones de la ville, représentant ainsi un taux de 8.37 % du volume global, dû à la concentration importante de la population au niveau de ce pôle (à multiple vocation), représenté par les citées de : Cité Frères Ben Zarefa, EL Kherba inférieur, cités frères Boussouf et cité Said Ali Aouad et les établissements à caractère éducatif et administratif tel que l'hôpital, École de la sureté ;

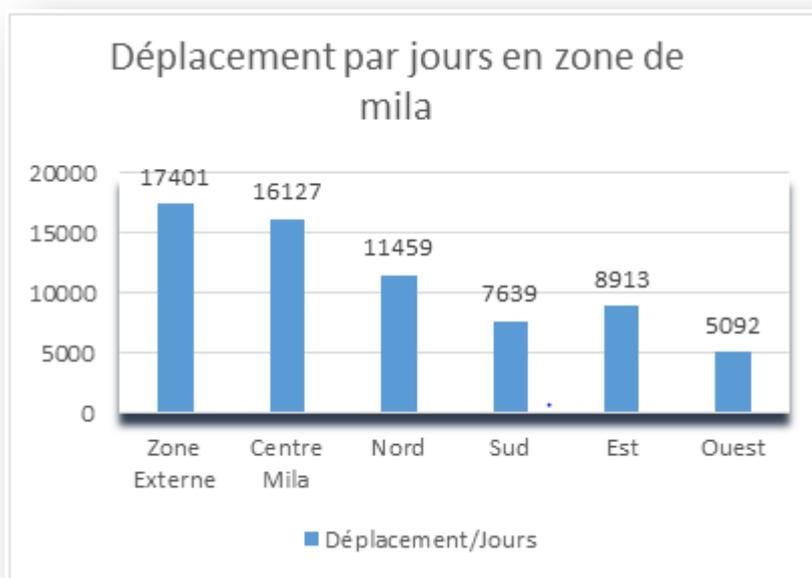


Figure 4.11: Histogramme de déplacement.

## Chapitre 5 : Tests et analyse des résultats

---

### 1. Introduction

Dans le présent chapitre on présente la démarche de test de TSC-PSO pour réguler le trafic routier via SUMO. Dans laquelle notre but c'est de trouver un programme des feux de signalisation de trafic plus efficace que les plans de feu existants.

### 2. Démarche de test

Nous effectuons une étude comparative entre le contrôle via TSC\_PSO et via feu fixe :

- Le contrôle via feux fixer est implémenté sans exécution de code python ; l'outil Netconvert de SUMO détermine les programmes cycliques des feux et les enregistre sur le fichier .net.
- Le contrôle via TSC-PSO est programmé sous python en interaction à SUMO. Nous avons conduit plusieurs essais en variant les coefficients d'accélération. Une fois la meilleure particule « gbest » de chaque version est déduite, c.à.d en fin des calculs du .py, elles sont soumises à la simulation pour extraire les traces et analyser les performances de chacune.

Après plusieurs essais et testes d'exécution des programmes avec des différentes valeurs des coefficients d'accélération pour TCS-PSO, on a obtenu les résultats suivants :

	C1	C2	C3	C4	DD+TA (TSC_PSO)	Itération
Exp1	2.02	2.15	1.5	1	174.81	10
Exp2	1	1.5	2.15	2.02	155.88	10
Exp3	2.02	2.15	2	1.9	151.2	2
Exp4	2.15	2.02	1.9	1	180.29	5
Exp5	2.02	2.15	2	2	163.53	8
Exp6	2.15	2.02	2.15	1	170.32	5
SUMO	/	/	/	/	157.55	/

Tableau 5.1: Résultats d'exécution de TSC\_PSO avec plusieurs valeurs différentes des coefficients d'accélération.

La dernière ligne représente l'exécution de SUMO pour des feux usuels sans contrôle supplémentaire en PSO, c.à.d sans besoin des valeurs des coefficients

Suite à ces essais d'exécutions, on a fixé les valeurs suivantes :  $c_1=2.02$ ,  $c_2=2.15$ ,  $c_3=2$ ,  $c_4=1.9$ .

Ces derniers résultats d'exécution sont sauvegardés dans des fichiers.xml ou fichiers traces de simulation utilisés pour créer des graphes comparatifs en utilisant SUMO, qui offre des outils pour convertir ces traces à des graphes.

```
<step time="985.00" loaded="951" inserted="920" running="457" waiting="20"  
<step time="986.00" loaded="951" inserted="921" running="458" waiting="20"  
<step time="987.00" loaded="951" inserted="921" running="457" waiting="21"  
<step time="988.00" loaded="951" inserted="921" running="457" waiting="21"
```

Figure 5.1: Exemple de fichier trace « summary ».

```
<tripinfo id="3" depart="3.00" departLane="506210829_0" departPos="5.10" departSpeed="0.00"  
  <emissions CO_abs="1330.198474" CO2_abs="56217.010578" HC_abs="7.371906" PMx_abs="1.0530"  
</tripinfo>  
<tripinfo id="9" depart="9.00" departLane="402167126_0" departPos="5.10" departSpeed="0.00"  
  <emissions CO_abs="3905.638427" CO2_abs="245145.221846" HC_abs="23.708646" PMx_abs="5.06"  
</tripinfo>
```

Figure 5.2: Exemple d'un fichier trace « tripinfo ».

### 3. Analyse comparative

Les fichiers trace relatifs aux meilleures particules de chaque version donnent, après plusieurs exécutions, les graphes suivants :

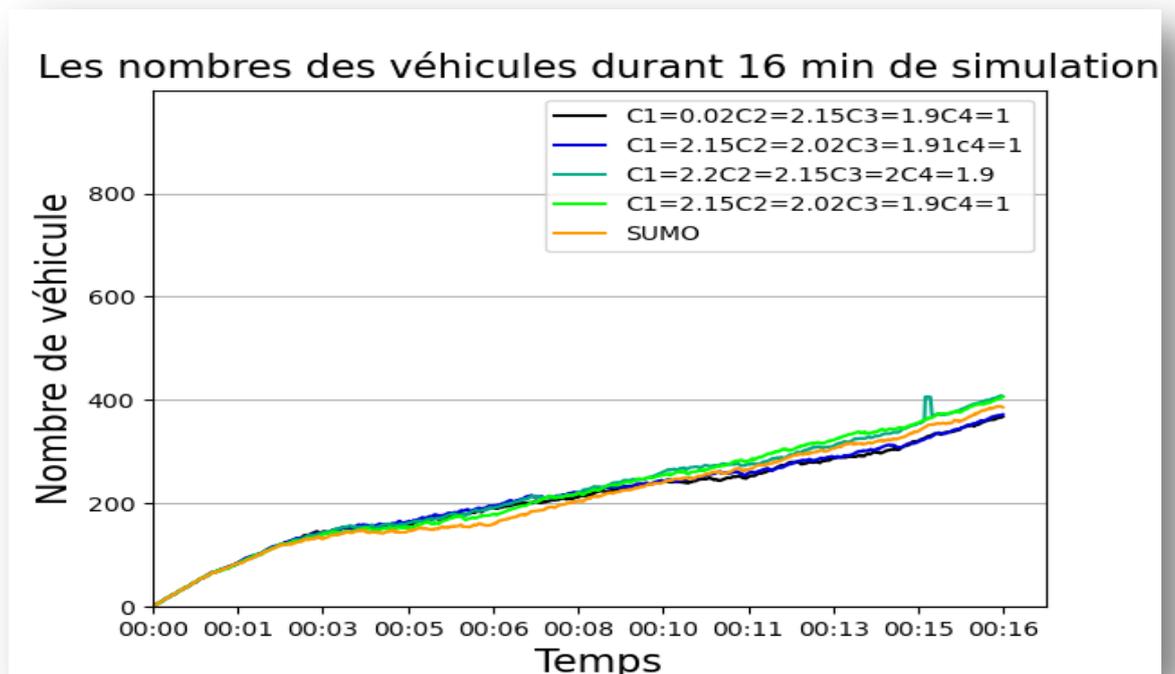


Figure 5.3: Comparaison du nombre de véhicules en déplacement sur le réseau.

Ce graphe montre le nombre de véhicules en déplacement à chaque instant de la simulation sur 16 minutes. Nous avons utilisé le même fichier qui commande les départs des véhicules pour toutes les expériences menées. Pourtant, le nombre est inférieur pour le feu fixe. Ceci est dû à la fonctionnalité de SUMO de ne pas éjecter des véhicules supplémentaires pour éviter la saturation au niveau entrée de zone étudiée.

On remarque que la version de PSO étudié améliore le nombre de véhicules par rapport au feu fixe : En effet, avec un feu fixe, seulement 380 véhicules sont en déplacement. Alors qu'avec TSC\_PSO en expérimentation 3 du tableau (vert foncé sur la figure), 410 véhicules atteignent leurs destinations.

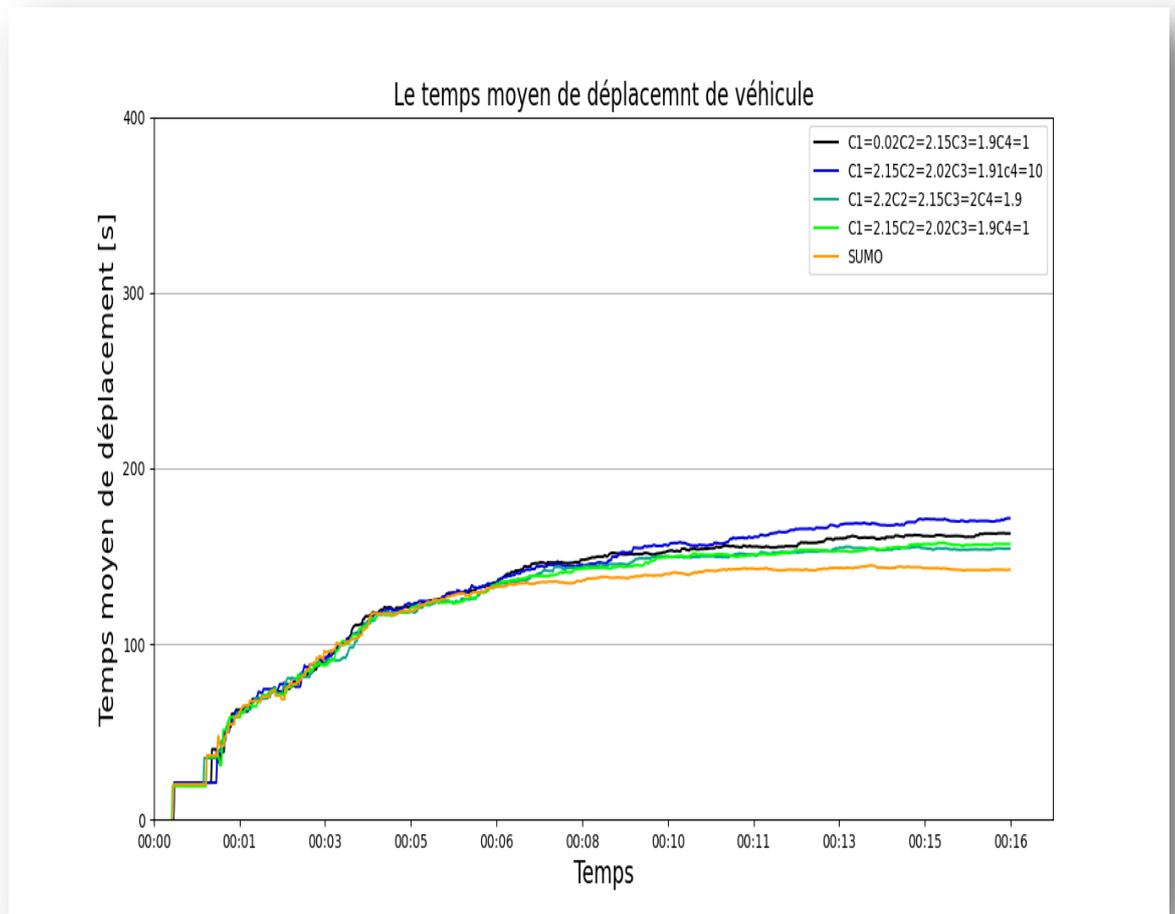


Figure 5.4: Comparaison de temps moyen de déplacement des véhicules.

Cette figure montre que le temps de déplacement moyen est plus intéressant via feu fixe que le contrôle en TSC-PSO. Pourtant, la fonction utilité indique clairement de meilleurs résultats pour PSO.

Ceci est dû au nombre de véhicules sur le réseau. En effet, on voit sur la figure précédente que le nombre de véhicules est plus important lorsqu'on travaille avec TSC-PSO. Nous notons de plus que la fonction utilité est la somme des durées de déplacement moyennes et des temps d'attentes moyennes. C'est pourquoi, nous présentons dans la figure suivante la deuxième composante de la fonction utilité : les temps d'attente moyens.

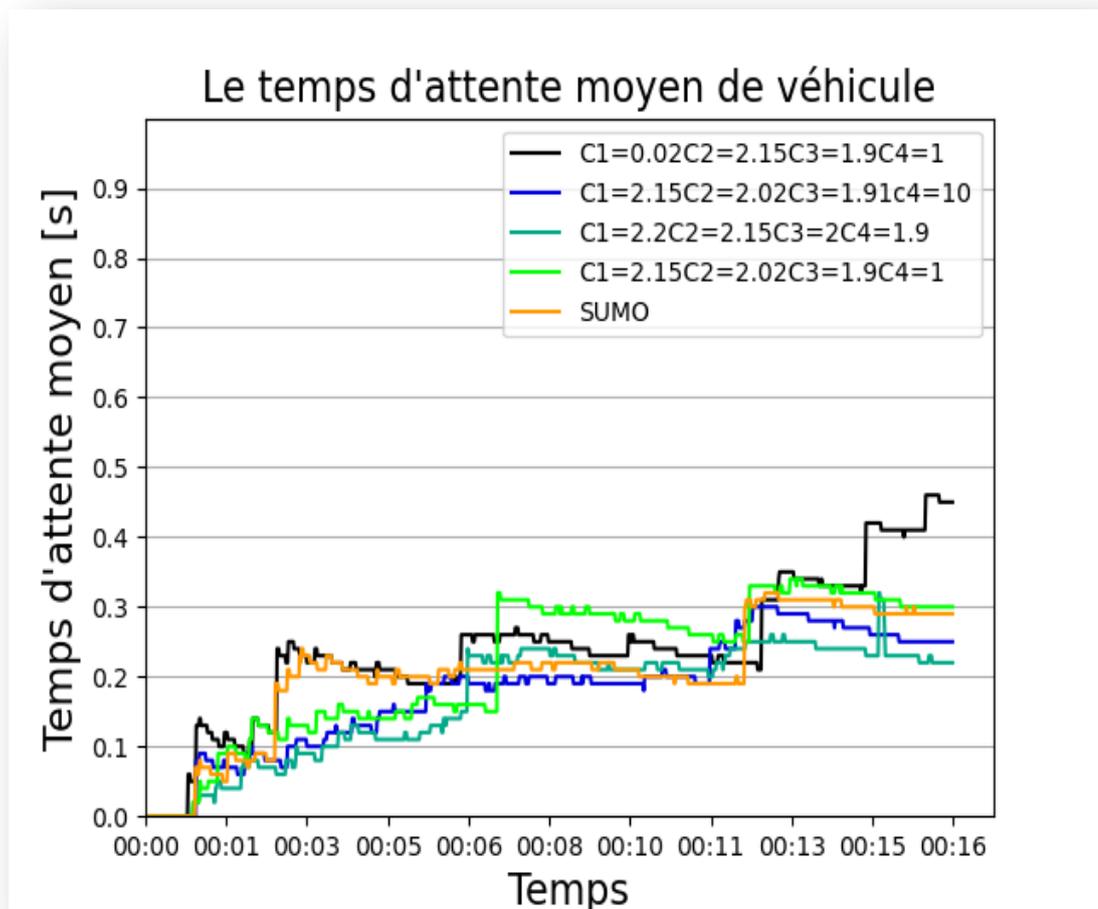


Figure 5.5: Comparaison des temps d'attente moyens de véhicule.

On a remarqué qu'effectivement, comme déduit de l'analyse sur la figure précédente, le contrôle en PSO est meilleur pour cette métrique. Ceci apparait plus clairement même lorsque le nombre de véhicules est plus important sur le réseau routier.

#### 4. Conclusion

Dans ce chapitre et dans le but de tester les performances et la robustesse d'algorithme de TSC-PSO étudié, et après les résultats qu'on a obtenu et qu'on a expliqué dans ce chapitre, on a trouvé que TSC\_PSO donne, dès les premiers essais, des résultats prometteurs. La conduite de tests supplémentaires en variant les métriques de l'algorithme mèneraient à des résultats meilleurs.

# Conclusion général

---

L'optimisation par trois essais particulières est un algorithme méta-heuristique parmi des différentes versions de PSO qui donne des résultats acceptables suite à la modification de quelques paramètres.

La première partie du manuscrit porte sur des généralités sur les différents problèmes d'optimisations et comment on peut les résoudre. Elle aborde les notions de trafic routier et de feu de signalisation ainsi qu'une présentation du simulateur utilisé. Pour la simulation on a choisi le simulateur SUMO pour évaluer la valeur de notre fonction objectif qui minimiser le temps d'attente et le temps de déplacement des véhicules.

La deuxième partie présente la démarche de traitement du problème de régulation du trafic routier via TSC-PSO sous SUMO. On commence par la modélisation du problème par sa fonction utilité et ses contraintes. L'interaction entre les variables du problème est assurée par simulation via SUMO sur une zone de la ville de MILA. L'objectif visé est de minimiser à la fois les temps d'attente et les durées de déplacement. Nous détaillons le principe et les paramètres du TSC-PSO implémenté pour réguler le trafic routier au niveau des feux de signalisation.

Le test de TSC\_PSO sur notre réseau, a abouti à diverses traces et diagrammes via SUMO. Les résultats analysés sont favorables à l'utilisation de cet algorithme pour déduire les meilleurs programmes des feux. Des essais plus nombreux relatifs à la variation des paramètres sont nécessaires.

# Bibliographie

---

- [1]. optimisation-multiobjectif-et-problemes-doptimisation-mono-objectifs. *wikimemoires*. [En ligne] [Citation : 19 06 2021.] <https://wikimemoires.net/2014/02/optimisation-multiobjectif-et-problemes-doptimisation-mono-objectifs>.
- [2]. **Belhous Roumaïssa, Boukezzoula Chaima**. *Simulation du contrôle du trafic routier via SUMO(Simulation of Urban Mobility)*. Mémoire de Master. Mila : Centre Universitaire AbdElhafidBoussouf Mila, 2019/2020.
- [3]. **Dor, Abbas El**. *Perfectionnement des algorithmes d'optimisation par essaim particulière : applications en segmentation d'images et en électronique*. paris : Université Paris-Est, 2012.
- [4]. **Rabie, BELLOUMI**. *Optimisation par Essaim de Particules Application à un système complexe*. Mémoire de MAGISTER. Université Batna 2 : s.n., 2015.
- [5]. *The Optimization Design of Urban Traffic Signal Control Based on Three Swarms Cooperative-particle Swarm Optimization*. **Yongsheng QIAN, Chunlei WANG, Hailong WANG, Zhetang WANG**. s.l. : School of Traffic and Transportation Engineering Lanzhou Jiaotong University Lanzhou, Gansu Province, China, 2007.
- [6]. **Fei Yan**. Contribution à la Modélisation et à la Régulation du Trafic aux Intersections , Intégration des Communications Véhicule-Infrastructure. Université de Technologie de Belfort-Montbéliard. 12 Mars 2012.
- [7]. **Lesort Ch ,Jean B**. Comprndre de Trafic Routier Méthodes et Calculs . Université Gustave Eiffel . Janvier 2010.
- [8]. **TOUHBI, Saâd**. *Élaboration d'un modèle multi-agents pour la génération*. THÈSE DE DOCTORAT. Marrakech , UNIVERSITÉ PIERRE ET MARIE CURIE & L'UNIVERSITÉ CADI AYYAD, 2018.
- [9]. **Layachi, Ilyes**. Le Trafic Routier. université de Annaba . 2017.
- Intersections , Intégration des Communications Véhicule-Infrastructure. Université de Technologie de Belfort-Montbéliard. 12 Mars 2012.
- [10]. **Babgai R**. Etude et Simulation des Feux de Circulation Cas de Carrefour Poste Centrale de Yaounde. Université de Yaounde 1. 2019.
- [11]. **Jean M** . Contribution a la Detection et a L'analyse d'Anomalies de Trafic en Milieu Urbain Regards sur l'Existant en Regulation Trafic Urbain . Université de Nancy 1 . 5 février 1993..
- [12]. **Cohen S**. Ingénierie du trafic routier - Presses des Ponts et Chaussées, Paris, 1990.
- Éditeur: Presses de l'École nationale des ponts et chaussées.

[13]. **Faye, Sébastien.** *Contrôle du trafic routier urbain par un réseau fixe de capteurs sans fil.* *Rapport Technique.* Télécom ParisTech, 2012.

[14]. **ASSAM, Amar et eddin, MOKDAD Mohamed dhia.** *Une approche multi-agents pour la simulation de la.* *Mémoire de Master02.* UNIVERSITE AKLI MOAND OULHADJE-BOUIRA, 2018/2019.

[15]. **ADIGUN, ADETUNDJI.** *GESTION DE L'ANONYMAT ET DE LA TRAÇABILITÉ DANS LES RÉSEAUX.* QUÉBEC , 2014.

[16]. **YEDDER, HANENE BEN.** *GESTION ADAPTATIVE DES VÉHICULES D'URGENCEUTILISANT DES INFORMATIONS D'URGENCEUTILISANT DES INFORMATIONS.* *MÉMOIRE DE MAÎTRISE.* UNIVERSITE DU QUEBEC EN OUTAOUAIS, 2014.

[17]. *Optimal Cycle Program of Traffic Lights With Particle Swarm Optimization.* **Jose Garcia-Nieto, Ana Carolina Olivera, Enrique Alba.** NO. 6, s.l. : IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, DECEMBER 2013, Vol. VOL. 17.