

# Filière Systèmes industriels

Orientation Power & Control

## Travail de bachelor Diplôme 2018

*Mariatharsan Arumugam*

*Mobilité flexible*

- *Professeur*  
Gilbert Morand
- *Expert*  
Philippe Schwery
- *Date de la remise du rapport*  
17.08.2018

Ce rapport est l'original remis par l'étudiant.  
Il n'a pas été corrigé et peut donc contenir des inexactitudes ou des erreurs.

Filière / Studiengang <b>SYND</b>	Année académique / Studienjahr <b>2017/18</b>	No TD / Nr. DA <b>pc/2018/45</b>
Mandant / Auftraggeber <input type="checkbox"/> HES—SO Valais <input type="checkbox"/> Industrie <input checked="" type="checkbox"/> Etablissement partenaire <i>Partnerinstitution</i> OFT (Office Fédéral des Transports)	Etudiant / Student <b>Mariatharsan Arumugam</b> Professeur / Dozent <b>Gilbert Morand</b>	Lieu d'exécution / Ausführungsort <input checked="" type="checkbox"/> HES—SO Valais <input type="checkbox"/> Industrie <input type="checkbox"/> Etablissement partenaire <i>Partnerinstitution</i>
Travail confidentiel / vertrauliche Arbeit <input type="checkbox"/> oui / ja <sup>1</sup> <input checked="" type="checkbox"/> non / nein	Expert / Experte <b>Philippe Schwery</b> Service de la mobilité Section IRT – Routes et trafic, Rue des Creusets 5, 1950 Sion	

Titre / Titel <b>Mobilité flexible</b>
Description / Beschreibung L'utilisation de véhicules électriques répondra certainement aux besoins de mobilité et changera, à coup sûr et de manière fondamentale, l'approvisionnement en électricité en termes d'énergie et de puissance dans les villes. A l'heure actuelle, les transports publics suivent un parcours bien défini en respectant scrupuleusement un horaire préétabli. Une piste consiste à "flexibiliser la mobilité", c'est-à-dire à adapter les parcours des véhicules électriques en fonction : – des demandes des utilisateurs de se déplacer d'un point A vers un point B, – des emplacements de la flotte de véhicules électriques, – du nombre de places libres dans les véhicules, – de l'état de charge des batteries de chacun des véhicules – de l'autonomie restante de chaque bus, et de la distance jusqu'au point de recharge (biberonage ou bornes spécifiques de recharge) L'objectif de ce travail est de mettre en place un système d'optimisation (algorithmes évolutionnaires) qui puisse répondre aux interrogations que l'on a sur le déploiement d'un tel système. Une connaissance des performances de charge et décharge, de batteries ou de « super-cap » embarquées dans les véhicules électriques est indispensable pour apporter des réponses fiables à la "mobilité flexible".

Signature ou visa / Unterschrift oder Visum Responsable de l'orientation / filière <i>Leiter der Vertiefungsrichtung / Studiengang:</i>  <sup>1</sup> Etudiant / Student : 	Délais / Termine Attribution du thème / Ausgabe des Auftrags: <b>16.05.2018</b> Présentation intermédiaire / Zwischenpräsentation <b>14 – 15.06.2018</b> Remise du rapport / Abgabe des Schlussberichts: <b>17.08.2018, 12:00</b> Expositions / Ausstellungen der Diplomarbeiten: <b>29, 30 – 31.08.2018</b> Défense orale / Mündliche Verfechtung: <b>04, 05 – 06.09.2018</b>
--	--

<sup>1</sup> Par sa signature, l'étudiant-e s'engage à respecter strictement la directive DI.1.2.02.07 liée au travail de diplôme.  
Durch seine Unterschrift verpflichtet sich der/die Student/in, sich an die Richtlinie DI.1.2.02.07 der Diplomarbeit zu halten.





## Mobilité flexible

Diplômant/e Mariatharsan Arumugam

### Objectif du projet

Développer un outil informatique d'aide à la décision qui joue des scénarios liés à la mobilité. Le logiciel met en œuvre des techniques dites de « Monte-Carlo » pour générer en sortie un tableau de bord groupant divers indicateurs de performance de mobilité.

### Méthodes | Expériences | Résultats

Afin d'optimiser la mobilité des transports publics, il faut connaître à l'avance les intentions et les habitudes des usagers. À partir de cela, le type de véhicule et le parcours idéal peuvent être adaptés selon les besoins, pour satisfaire à la fois les exploitants et les occupants, tout en réalisant des économies et en réduisant la pollution due au transport.

La méthode de Monte-Carlo, qui est utilisé lors de la simulation, désigne une famille de méthode algorithmiques visant à calculer une valeur numérique approchée en utilisant « des procédés aléatoires ». Les paramètres utilisés par le simulateur sont organisés hiérarchiquement dans une base donnée regroupant plus de 7 tables. Le logiciel de simulation réalisé, déplace virtuellement les individus d'un point A à un point B et suit le parcours de chacun des bus de la flotte qui dessert une région. Une fonction récursive permet de déterminer, pour tous les usagers d'une région, le chemin qui minimise la durée de déplacement en transport public tout en répondant à leur besoin en mobilité.

Les résultats de la simulation sont visibles dans un tableau de bord où sont contenus plusieurs indicateurs qui vont aider les décideurs des transports publics à répondre aux interrogations de la mobilité.

### Travail de diplôme | édition 2018 |

Filière  
*Systèmes industriels*

Domaine d'application  
*Power & Control*

Professeur responsable  
*Dr. Gilbert Morand*  
*gilbert.morand@hevs.ch*

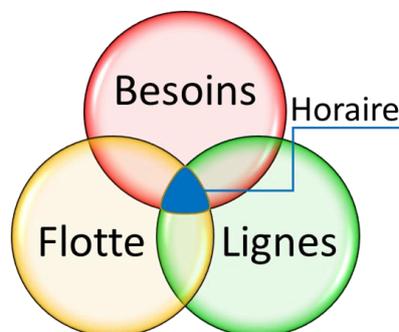
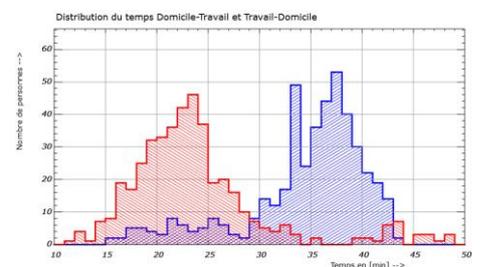


Diagramme de Venne : Lorsque les trois paramètres principaux de la mobilité sont réunis alors un horaire idéal est créé.



Exemple de plot du tableau de bord : Il s'agit ici d'un indicateur social qui montre la distance que les individus parcourent en bus.



*« Rien ne sert de courir, il faut prendre le bus »*

*Arumugam Mariatharsan*

Je tiens à remercier toutes les personnes qui ont contribuées de près ou de loin à la réalisation de ce projet :

- ❖ M. Gilbert-André Morand, professeur à la HES-SO, qui a su parfaitement m'épauler tout au long du projet avec de la passion et en faisant preuve d'une grande disponibilité. De plus, il m'a mis à disposition différentes librairies qui ont été réalisées par ses soins et dont j'ai eu l'utilité pour le bon fonctionnement de mon projet.
- ❖ Mes collègues de bureau Thomas Moser, Loïc Gillioz, Joost Laros et Mauro Stephan pour leur avis et conseils tout au long du projet.
- ❖ Florian Smets qui a participé et qui m'a aidé lors des séances avec le professeur.

# Table des matières

Liste des figures .....	i
Liste des tableaux .....	i
Liste des fonctions .....	iv
Glossaire .....	v
1. Introduction .....	1
1.1 Contexte .....	1
1.2 Objectif .....	2
1.3 Cahier des charges .....	2
1.4 Logiciels utilisés .....	2
2. Développement de la simulation .....	3
2.1 Méthode Monte-Carlo .....	3
2.1.1 Définition .....	3
2.1.2 Exemple .....	3
2.1.3 Intégration au projet .....	4
2.1.4 Loi normale .....	4
2.1.5 Distributions quelconques .....	4
2.2 Simulation .....	6
2.2.1 Définition .....	6
2.2.2 Quand utiliser un simulateur ? .....	6
2.2.3 Avantages de la simulation .....	6
2.2.4 Inconvénients de la simulation .....	6
2.3 Concept de la simulation .....	6
2.3.1 Les entrées de la simulation (Input) .....	7
2.3.2 Simulation effectuée .....	8
2.3.3 Les sorties de la simulation (Output) .....	8
2.4 Méthode de simulation .....	8
2.4.1 Première distinction : .....	8
2.4.2 Deuxième distinction : .....	8
2.4.3 Troisième distinction : .....	9

2.5	Développement d'un modèle de simulation.....	9
2.5.1	Identification du problème.....	9
2.5.2	Formulation du problème .....	9
2.6	Description des modèles choisis .....	10
2.6.1	Energétique .....	10
2.6.2	Environnementale .....	10
2.6.3	Economique.....	10
2.6.4	Social.....	10
3.	Fonctionnement de la simulation.....	11
3.1	Listes mémoire .....	11
3.2	Evènement et Timer .....	12
3.3	Résumé de la simulation .....	14
3.4	Les tables de données à fournir en entrée de la simulation .....	17
3.4.1	Table Lieu_Home.....	17
3.4.2	Table Lieu_Work.....	18
3.4.3	Table flotte .....	18
3.4.4	Table Arrêt.....	18
3.4.5	Table Trajets .....	19
3.4.6	Table Individus.....	20
3.4.7	Table Besoins.....	20
3.4.8	Table Engagements .....	21
3.5	Les tables de données générées par le logiciel de simulation .....	22
3.5.1	Table Horaire .....	22
3.5.2	Table Déplacement.....	25
3.6	Fonctions importantes programmées.....	30
3.6.1	Gestions du temps.....	30
3.6.2	Calcul de distance .....	31
3.6.3	Chemin idéal.....	32
3.7	Autres fonctions programmées.....	38
3.8	Fonctions pour les plots .....	38
4.	Test et Résultats .....	39

4.1	Test.....	39
4.1.1	Situation et conditions des tests .....	40
4.2	Résultats.....	40
4.2.1	Arrêts.....	41
4.2.2	Parcours.....	42
4.2.3	Distance de déplacement.....	42
4.2.4	Temps de déplacement .....	43
5.	Bus électrique.....	44
6.	Améliorations futures.....	46
7.	Conclusion .....	47
8.	Bibliographie.....	48
9.	Annexes .....	49
9.1	Annexe 1 : Schéma relationnel des tables de données.....	49
9.2	Annexe 2 : Schéma déplacement pour 0 correspondance.....	50
9.3	Annexe 3 : Schéma de déplacement pour 1 correspondance.....	51
9.4	Annexe 4 : Ligne 23 TOSA.....	52
9.5	Annexe 5 : Informations du Bus TOSA.....	53
9.6	Annexe 6 : Parcours des navettes autonomes .....	54
9.7	Annexe 7 : Informations du nombre de voyageurs.....	55
9.8	Annexe 8 : Représentation sur Power Map .....	55
9.9	Annexe 9 : Installation module xlsx.....	56
9.10	Annexe 10 : Mise à jour de l'interface utilisateur .....	57

## Liste des figures

Figure 1 : Lac avec le rectangle autour.....	3
Figure 2 : Lac avec la zone autour après le lancer de pierre .....	3
Figure 3 : Distribution de Gauss .....	4
Figure 4 : Distribution de la ponctualité.....	5
Figure 5 : Distribution de la distance Domicile – Arrêt .....	5
Figure 6 : schéma bloc de la simulation .....	6
Figure 7 : Diagramme de Venn de la mobilité.....	7
Figure 8 : Concept d'un modèle .....	9
Figure 9 : Modèle Energétique de la mobilité.....	10
Figure 10 : Machine d'état .....	12
Figure 11 : Logique de simulation .....	13
Figure 12 : Définition de la période de simulation.....	14
Figure 13 : L'individu 0 se trouve à son domicile (départ) .....	14
Figure 14 : l'individu 0 se rend à l'arrêt 2 .....	15
Figure 15 : l'individu 0 attend le bus .....	15
Figure 16 : Les bus passent par l'arrêt 2.....	15
Figure 17 : L'individu 0 se trouve dans le bus 3.....	15
Figure 18 : l'individu 0 est arrivé à son arrêt de destination.....	16
Figure 19 : l'individu 0 est arrivé à son lieu de travail (Destination) .....	16
Figure 20 : Schéma relationnel entre les tables .....	17
Figure 21 : Heure arrivée réelle du bus en retard .....	23
Figure 22 : Heure arrivée réelle du bus en avance.....	23
Figure 23 : Horaire planifié et réel .....	24
Figure 24 : Schéma déplacement avec 0 correspondance .....	28
Figure 25 : Lancement du Timer pour le déplacement jusqu'à l'arrêt.....	29
Figure 26 : Schéma déplacement avec 1 correspondance .....	29
Figure 27 : Structure MON_TEMPS .....	30
Figure 28 : Définition de la latitude et de la longitude.....	31
Figure 29 : Exemple de récursivité .....	33
Figure 30 : Exemple de chemins.....	34
Figure 31 : Chemin N°0.....	34
Figure 32 : Chemin N°1.....	34

Figure 33 : Chemin N°2.....	34
Figure 34 : Chemin N°3.....	35
Figure 35 : Logique pour trouver l'horaire idéal .....	36
Figure 36 : Interface utilisateur .....	39
Figure 37 : Tableau de bord.....	39
Figure 38 : Carte de la zone de test avec les lignes utilisées.....	40
Figure 39 : Histogramme de la distance jusqu'à l'arrêt.....	41
Figure 40 : Parcours du Scénario N°1 .....	42
Figure 41 : Parcours du Scénario N°2 .....	42
Figure 42 : Distance parcourue par les individus pour le Scénario N°1 .....	42
Figure 43 : Distance parcourue par les individus pour le Scénario N°2 .....	42
Figure 44 : Temps passés par les individus dans le bus pour le Scénario N°1.....	43
Figure 45 : Temps passés par les individus dans le bus pour le Scénario N°2.....	43
Figure 46 : TL Transports publics de la région lausannoise SA.....	44
Figure 47 : Recharge par biberonnage Bus TOSA.....	45
Figure 48 : Navette autonome du projet « SmartShuttle » .....	45

## Liste des tableaux

Tableau 1 : Cahier des charges .....	2
Tableau 2 : Table Lieu_Home .....	17
Tableau 3 : Table Lieu_Work .....	18
Tableau 4 : Table Flotte .....	18
Tableau 5 : Table Arrêts.....	18
Tableau 6 : Table Trajets .....	19
Tableau 7 : Table Individus .....	20
Tableau 8 : Table Besoin.....	20
Tableau 9 : Exemples de profil .....	21
Tableau 10 : Table Engagements.....	21
Tableau 11 : Table horaire en cours de création.....	22
Tableau 12 : Statuts de l'horaire du bus.....	24
Tableau 13 : Partie de la table Horaire finale.....	25
Tableau 14 : Table Déplacements Partie 1 .....	25
Tableau 15 : Table Déplacement Partie 2 .....	26
Tableau 16 : Table Déplacement Partie 3 .....	27
Tableau 17 : Exemple de liste déplacement individu.....	28
Tableau 18 : Activité de l'individu .....	30
Tableau 19 : Calcul de la distance .....	32
Tableau 20 : Différentes possibilités de chemins.....	33
Tableau 21 : Liste de parcours du chemin N°2 .....	35
Tableau 22 : Liste de parcours du chemin N° 3 .....	35
Tableau 23 : Liste infos parcours.....	36
Tableau 24 : Liste correspondance.....	37
Tableau 25 : Liste correspondance.....	37

## Liste des fonctions

Fonction 1 : Fonctions utilisées pour les listes mémoires.....	11
Fonction 2 : Fonctions complémentaires créées pour les listes mémoires .....	11
Fonction 3 : GetTempsArrivee.....	22
Fonction 4 : GetNextiHo .....	23
Fonction 5 : ReturnChance .....	26
Fonction 6 : ChercheArretPlusProche .....	26
Fonction 7 : Determine_t70Horaire .....	31
Fonction 8 : From_t70_to_Mon_Temps .....	31
Fonction 9 : AssigneStruct_MON_TEMPS .....	31
Fonction 10 : ComputeDist.....	32
Fonction 11 : DetermineChemins.....	33
Fonction 12 : DeterminePortionChemin .....	33
Fonction 13 : Presence .....	33
Fonction 14 : ComputeTempsParcours .....	36
Fonction 15 : GetCorrespondance .....	36
Fonction 16 : GetTempsTotal .....	37
Fonction 17 : GetTrajetsPlusCourt .....	37
Fonction 18 : hbook1_.....	38
Fonction 19 : hfill_.....	38

## Glossaire

Mots	Définition
<b>Individu</b>	Une personne qui utilise les transports
<b>Usager</b>	Une personne qui utilise les transports
<b>Home</b>	Le domicile des individus
<b>Work</b>	Tous les lieux différents du domicile où les individus peuvent se rendre
<b>Timer</b>	Une minuterie qui est lancé pour savoir quand un évènement doit se produire
<b>Arrêt</b>	Il s'agit d'un arrêt de bus
<b>US</b>	Abréviation utilisée pour usager
<b>Lieu de travail</b>	Lieux où les personnes se rendent en prenant le bus.
<b>Utilisateur</b>	Personne qui utilise le logiciel de simulation
<b>Ligne</b>	Ligne de bus ou ligne du tableau, dépend le contexte dans lequel il est utilisé

# 1. Introduction

## 1.1 Contexte

Depuis la nuit des temps, la mobilité est l'un des aspects principaux reflétant la capacité humaine à évoluer et interagir avec son environnement. La révolution industrielle, au début du XIX<sup>e</sup> siècle, apporte avec elle un développement discontinu des transports, des technologies employées et du maillage formant le réseau routier.

Aujourd'hui, la mobilité fait partie intégrante de la vie de l'homme quelles que soient les activités du quotidien, comme les déplacements journaliers et répétés pour se rendre dans des établissements publics, au travail ou à l'école ou à plus grande échelle pour des trajets intercontinentaux. La qualité et la richesse des transports reflètent l'état de santé d'une société.

Cependant, force est de constater que les déplacements des voyageurs restent encore complexes à maîtriser. Trajets, horaires, nombre d'individus sont autant de paramètres qui peuvent, lorsque les prévisions sont incertaines, tendre vers des événements préoccupants comme les heures de pointe où la majorité des occupants empruntent les transports publics et privés.

À l'heure actuelle, l'intérêt des voyageurs est de réduire le temps passé dans les transports plutôt que la distance parcourue. Avec cela, ils portent une importance particulière à deux caractéristiques que sont le confort et la sécurité.

En Valais, les transports publics, composés uniquement de bus et de cars postaux, suivent un trajet prédéfini qui compte un nombre important d'arrêts consécutifs mais qui sont facultatifs. Parfois, un détour doit être effectué pour satisfaire les voyageurs habitants dans des régions reculées. Des tronçons sont empruntés plusieurs fois par jour alors que les arrêts qui s'y trouvent ne sont pas marqués. Il s'agit alors d'un problème d'optimisation car un itinéraire plus court pourrait être adopté selon les besoins.

En effet, les chauffeurs des transports publics n'ont, à ce jour, aucun moyen de connaître la destination des usagers jusqu'à ce qu'ils demandent l'arrêt ou encore si des personnes attendent le bus le long de la ligne qu'ils vont parcourir. En adaptant le trajet du bus selon les besoins, un gain dans les domaines énergétique et financier peut être réalisé.

Les transports publics ont à leur disposition plusieurs véhicules destinés au transport de personnes. Pendant les heures d'affluence, les véhicules disposant d'un maximum de place sont les plus appropriés. Mais le reste du temps, il est possible d'utiliser des véhicules plus économiques et plus petits car la demande n'est pas aussi importante. Cependant, il faut connaître en amont le nombre de personnes à déplacer pour pouvoir adapter le type de véhicule à utiliser.

Afin d'améliorer les transports publics en répondant à une demande qui ne cesse de croître (Annexe 7) et en prenant en compte la croissance démographique et les changements climatiques, il est plus que nécessaire de connaître à l'avance les intentions et les habitudes des usagers. De ce constat, le type de véhicule et le parcours idéal peuvent être adaptés selon les besoins pour que les exploitants et les occupants soient satisfaits, tout en réalisant des économies et en réduisant la pollution due au transport.

## 1.2 Objectif

Un outil d'aide à la décision doit être développé et réalisé. Il a pour but de jouer dynamiquement des scénarios liés à la mobilité. Le logiciel mettra en œuvre des techniques dites de « Monte-Carlo » pour générer en sortie un tableau de bord groupant divers indicateurs de performance de mobilité.

## 1.3 Cahier des charges

Voici le cahier des charges établi à partir de la donnée reçue :

Repères	Description	Critères d'appréciation	Contraintes	Niveau d'appréciation
FP1	Développer un concept de simulation de la mobilité	> Adaptable pour d'autres villes > Choix du scénario > Comparaison des scénarios	> Répondre aux besoins de la mobilité > Respecter les caractéristiques des véhicules	> Proche à 80 % de la réalité > Pas d'incohérences scénaristiques
FP2	Réaliser un outil intégrant le concept			
FP3	Intégrer un tableau de bord contenant des indicateurs liés à la mobilité	> Possibilité de choisir les indicateurs	> Créer des modèles concrets	> Résultat réaliste

Tableau 1 : Cahier des charges

## 1.4 Logiciels utilisés

Voici la liste des logiciels utilisés pour ce projet :

- Qt Creator (Langage C++)
- Suite Microsoft Office

## 2. Développement de la simulation

### 2.1 Méthode Monte-Carlo

Certains problèmes faisant intervenir un phénomène aléatoire peuvent être étudié de façon analytique. Cependant, la solution trouvée peut se révéler être trop complexe. Pour éviter ce genre de difficulté, la méthode Monte-Carlo est une alternative. La plupart du temps, les conditions du problème ne permettent pas de disposer de paramètres suffisamment réels. En effet, il est nécessaire de prendre en compte un certain nombre de situations réelles pour estimer les paramètres des différentes lois de probabilité.

#### 2.1.1 Définition

La méthode de Monte-Carlo<sup>1</sup> désigne toute méthode visant à estimer une quantité numérique qui utilise des nombres aléatoires, c'est-à-dire des techniques probabilistes. Stanislaw Ulam et John von Neumann l'appelèrent ainsi, en référence aux jeux de hasard pratiqués dans les casinos de Monte-Carlo, au cours du projet Manhattan qui produisit la première bombe atomique pendant la Seconde Guerre mondiale.

Les méthodes de Monte-Carlo<sup>2</sup> sont particulièrement utilisées pour calculer des intégrales en dimensions plus grandes que 1 (calculer des surfaces et des volumes). Elle présente le double avantage d'être simple d'utilisation et de pouvoir être appliquée à un très large éventail de problèmes. Elle peut être utilisée aussi bien en assurance pour évaluer le montant d'une prime qu'en physique comme ce fut le cas en 1945 lors du bombardement atomique d'Hiroshima et Nagasaki.

#### 2.1.2 Exemple

La détermination de la superficie d'un lac est un classique en vulgarisation de la méthode de Monte-Carlo. Soit un rectangle dont les longueurs sont connues. A l'intérieur de cette zone se trouve un lac dont la surface est inconnue car les courbes ne sont pas régulières.

Pour trouver l'aire du lac en connaissant celui du rectangle, il faut :

1. Lancer M pierres de manière aléatoire dans cette zone.
2. Compter le nombre N de pierres qui ont atterri sur le terrain.
3. Trouver le nombre de pierres K = M-N qui sont tombés dans le lac.

Ainsi la surface du lac =  $\frac{K}{M} \cdot \text{aire du rectangle}$

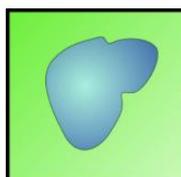


Figure 1 : Lac avec le rectangle autour

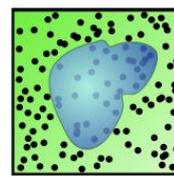


Figure 2 : Lac avec la zone autour après le lancer de pierre

<sup>1</sup> [https://interstices.info/jcms/int\\_69164/la-simulation-de-monte-carlo](https://interstices.info/jcms/int_69164/la-simulation-de-monte-carlo)

### 2.1.3 Intégration au projet

Lors de ce projet, cette méthode servira à calculer différentes entrées aléatoires, tels que le nombre de passagers, l'heure de passage du bus ou encore le temps de trajet. Ce dernier varie selon les circonstances du moment mais qui se situe toujours autour d'une valeur d'espérance avec un écart type.

Par exemple, pour déterminer le nombre de personnes qui prennent le bus le matin à un endroit précis, même en ayant connaissance du nombre de personne à l'avance, il peut y avoir des imprévus. Afin de simuler en prenant en compte tous ces événements aléatoires, il faut utiliser une distribution comme celle de Gauss.

### 2.1.4 Loi normale

Il repose sur l'estimation de deux paramètres de la population statistique :

- La moyenne  $\mu$
- L'écart type de  $\sigma$
- Variable  $x$  se situe dans la surface de

L'équation de la courbe de la loi normale est la suivante :

$$f(x) = \frac{1}{\sqrt{2 \cdot \pi \cdot \sigma^2}} \cdot e^{-\frac{(x-\mu)^2}{2 \cdot \sigma^2}}$$

Équation 1 : loi normale

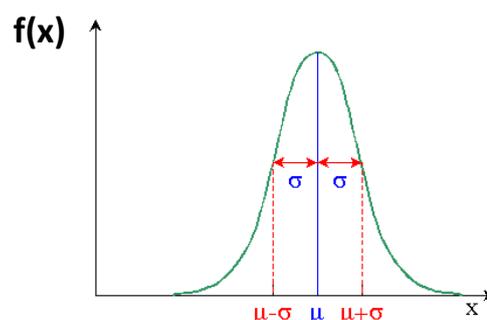


Figure 3 : Distribution de Gauss

### 2.1.5 Distributions quelconques

Le Monte-Carlo fonctionne également avec d'autres distributions que celle de Gauss. Par exemple, pour la ponctualité souhaitée des personnes lorsqu'ils arrivent à destination, une distribution a été faite à l'aide de la fonction hTir. (3.8 Fonctions pour les plots)

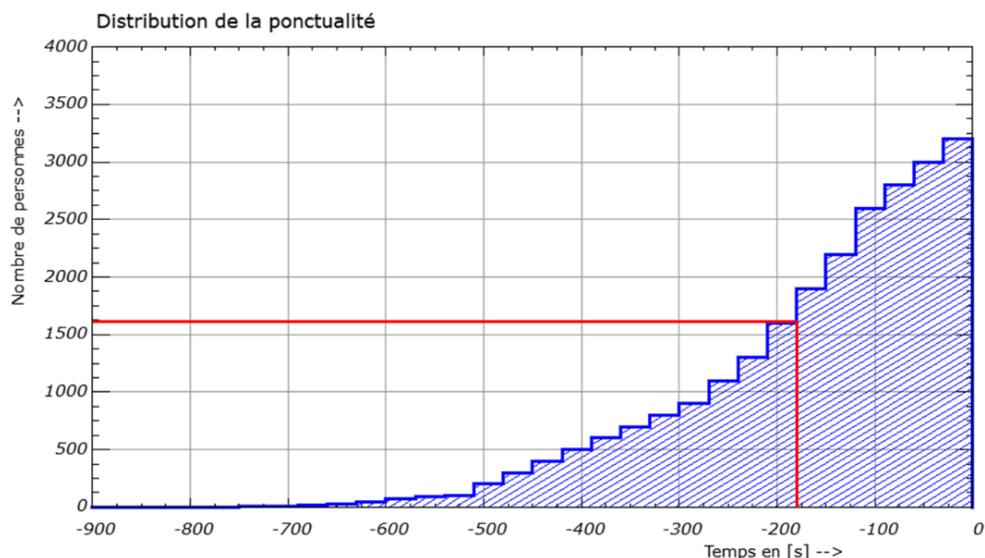


Figure 4 : Distribution de la ponctualité

La distribution de la figure ci-dessus est réalisée pour plus 35'000 personnes et montre que pour le tirage effectué, en rouge, il y a environ 1600 personnes qui arrivent aux alentours de 3 minutes avant l'heure prévue. La distribution a été réalisée en prenant en compte le fait que personne ne souhaite arriver en retard.

Évidemment cette distribution ne reflète pas la réalité, mais elle apporte un phénomène stochastique qui éloigne la simulation de la perfection mais qui la rapproche de la réalité. Une distribution réelle peut être réalisée par exemple pour la distance entre le domicile des individus et l'arrêt de bus qui se trouve le plus proche.

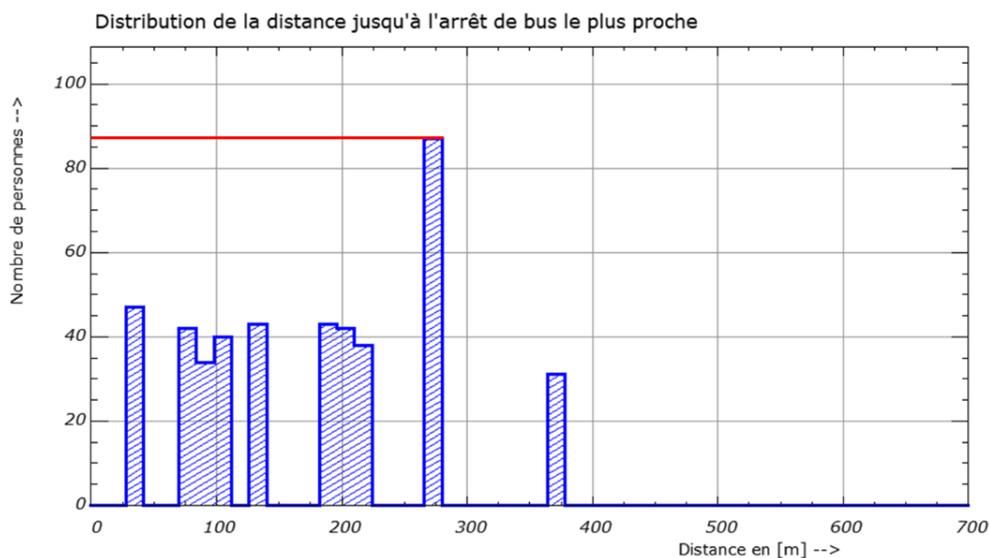


Figure 5 : Distribution de la distance Domicile – Arrêt

Cette seconde distribution a été réalisée pour 300 personnes dont les domiciles ont été tirés au sort parmi 30 habitations. Le constat est que pour les tables d'entrées utilisées ici, environ 87 personnes habitent dans un périmètre de 280 mètres autour d'un arrêt. Ce type de distribution reflète la réalité et dépend uniquement du nombre d'habitations et d'arrêts de bus qui sont donnés dans les tables.

## 2.2 Simulation

### 2.2.1 Définition

La simulation est une technique numérique pour mener des expériences à l'aide d'un ordinateur qui utilise des caractéristiques assurées ou stochastiques et qui implique l'utilisation de modèles mathématiques qui décrivent le comportement d'un système sur des périodes étendues en temps réel.

### 2.2.2 Quand utiliser un simulateur ?

- Lorsqu'il faut une visualisation globale.
- Lorsque l'expérience ne peut pas être menée dans la réalité.
- Lorsqu'un nombre important et différent de paramètres doivent être changés.

### 2.2.3 Avantages de la simulation

- Peut être expérimenté avec de nouveaux scénarios qui n'existent pas encore.
- La simulation donne le contrôle sur le temps .
- La simulation est moins chère que de faire des expériences
- La simulation est sans danger car elle est en « laboratoire».

### 2.2.4 Inconvénients de la simulation

- Les modèles de simulation exigent un nombre considérable de données d'entrées qui sont difficiles, voire parfois impossibles à obtenir.
- Les modèles de simulation doivent être le plus correcte possible sinon les décisions basées sur ces résultats auront des conséquences importantes.
- Le développement d'un modèle de simulation exige des connaissances dans plusieurs domaines tels que la programmation, les probabilités et les analyses des résultats obtenus.

## 2.3 Concept de la simulation

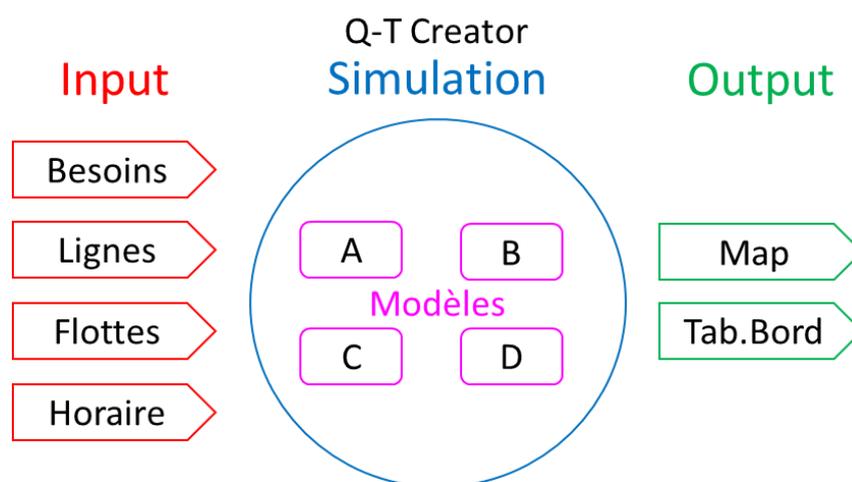


Figure 6 : schéma bloc de la simulation

### 2.3.1 Les entrées de la simulation (Input)

La simulation doit retracer la mobilité avec un scénario proche de la réalité. Pour cela, il faut déterminer les paramètres à prendre en compte pour représenter la mobilité le plus fidèlement possible.

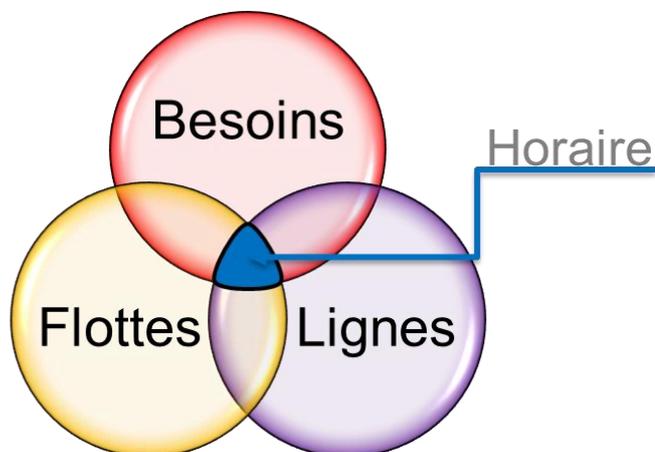


Figure 7 : Diagramme de Venn de la mobilité

Les paramètres principaux de la mobilité sont représentés dans le diagramme de Venn ci-dessus. L'utilisation de ce genre de représentation à l'avantage de montrer les relations logiques qui existent entre les différents ensembles.

- **Besoins :**

Le besoin en mobilité est quelque chose d'important pour chaque individu. Les paramètres à prendre en compte individuellement sont le lieu de départ (Home), le lieu de destination (Work) et le profil d'utilisateur. Ce dernier paramètre définit quand et combien de fois l'individu a besoin d'être déplacé par jour.

- **Lignes :**

La ligne est le chemin qu'emprunte les transports publics. Elle est constituée de différents arrêts qui se trouvent tout au long du parcours. Les paramètres à prendre en compte sont le temps et la distance entre chaque arrêt.

- **Flottes :**

La flotte de véhicule<sup>3</sup> est l'ensemble du parc automobile d'une entreprise. Elle regroupe tous les types de véhicules disponibles au bon fonctionnement de la société. La complexité de la gestion d'une flotte automobile ne permet plus une navigation à l'aveugle, car chaque décision a un impact sur le plan financier et environnemental. Les paramètres à prendre en compte sont le type de véhicule (moteur à essence, hybride ou électrique) et les caractéristiques (nombre de places, consommation) de ce dernier.

<sup>3</sup> <https://voitures-societe.ooreka.fr/comprendre/formule>

- **Horaire :**

L'horaire indique des heures où a lieu une activité. Dans cette simulation, il s'agit de l'heure de passage des transports publics. L'horaire idéal se trouve au centre de la relation des trois ensembles (Figure 7 *Erreur ! Source du renvoi introuvable.*). Selon l'heure de la journée, le profil de déplacement varie. Il faudra donc en tenir en compte dans les paramètres de l'horaire.

### 2.3.2 Simulation effectuée

La simulation reproduit le déplacement des individus et des transports publics sur une période donnée. En se basant sur les données du diagramme de Venn de la Figure 7, la simulation déplacera les individus d'un point A à un point B en passant par des arrêts qui se trouvent sur des trajets prédéfinis.

### 2.3.3 Les sorties de la simulation (Output)

En sortie, la simulation donnera des réponses sous deux formes. La première est dans un tableau de bord où selon le choix des indicateurs, différents graphiques apporteront des réponses précises. La seconde sera sur un plan où les informations telles que la position des véhicules et des personnes seront affichées en fonction de l'heure et du lieu.

## 2.4 Méthode de simulation

### 2.4.1 Première distinction :

Quasiment tous les modèles de simulation de la mobilité sont dynamiques et leur principale variable indépendante est le temps. Cependant, il faut différencier le modèle continu qui s'oppose au modèle discret.

- **Continu** : Le temps est découpé en tranches égales sans que l'on puisse distinguer des événements. L'analyse se fait chaque seconde, chaque minute, etc.
- **Discret** : un événement doit se produire pour que l'on puisse observer un changement. Par exemple, si personne ne se présente à un arrêt ou que le bus ne s'y arrête pas, il n'y a pas d'évènement et le système ne change pas, donc il n'y a rien à observer.

C'est le modèle **discret** qui sera utilisé lors de ce projet.

### 2.4.2 Deuxième distinction :

La simulation peut être classer selon le niveau de détails qu'il offre. Ainsi, il existe trois principaux degrés de précision :

- **Microscopique** : Il offre un haut niveau de détails pour les objets du système avec leurs interactions. Il prend plus de temps pour tout simuler.
- **Mésoscopique** : Il offre également un haut niveau de détails pour les objets du système, cependant leurs interactions sont moins précises.
- **Macroscopique** : Il représente à un faible niveau le détail des objets ainsi que leurs interactions.

Le niveau de détail sera pour l'instant le **mésoscopique**. Une fois le simulateur au point, le niveau de détails pourra être amélioré si le temps de simulation n'est pas trop important.

### 2.4.3 Troisième distinction :

Cette dernière classification différencie les modèles de simulation par les types de variables utilisées. Il existe deux modèles :

- **Déterministe** : Aucune variable n'est aléatoire. Elles ont toutes une relation de type mathématique, statistique ou logique.
- **Stochastique** : Des fonctions de probabilités sont utilisées. Certaines variables sont aléatoires.

## 2.5 Développement d'un modèle de simulation

### 2.5.1 Identification du problème

Il faut définir le problème explicitement :

- Quelles entrées ont un impact sur les sorties désirées ?
- Quels éléments perturbateurs sont à prendre en compte ?
- Quelles sont les limites temporelles et spatiales du problème ?
- Quels sont les éléments stochastiques du problème s'il y en a ?

### 2.5.2 Formulation du problème

Le graphique ci-dessous représente les trois parties qui composent un modèle :

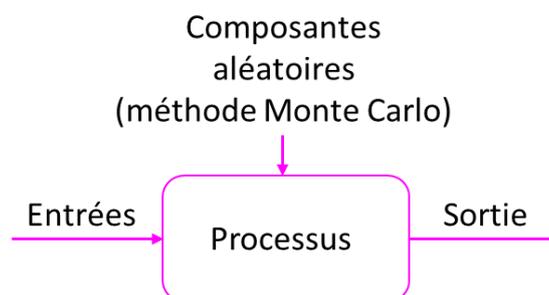


Figure 8 : Concept d'un modèle

Le but de ce concept est d'identifier toutes les entrées dont a besoin le modèle et d'y associer les sorties concerner.

- Les entrées seront les paramètres du besoin, de la flotte, de la ligne et de l'horaire.
- Les composantes aléatoires sont les données qui seront obtenues par la méthode de Monte Carlo.
- Les sorties seront les indicateurs qui seront visibles dans le tableau de bord ainsi que sur la visualisation sur le plan.

## 2.6 Description des modèles choisis

Les quatre modèles mathématiques importants liée à la mobilité sont les suivants :

### 2.6.1 Energétique

Ce modèle concerne la consommation d'un bus. Que ce soit en L/100 km pour des bus Diesel et hybride ou en kW pour un bus électrique, il faut connaître la consommation du bus pour optimiser la mobilité. Voici un exemple de modèle pour l'énergie :

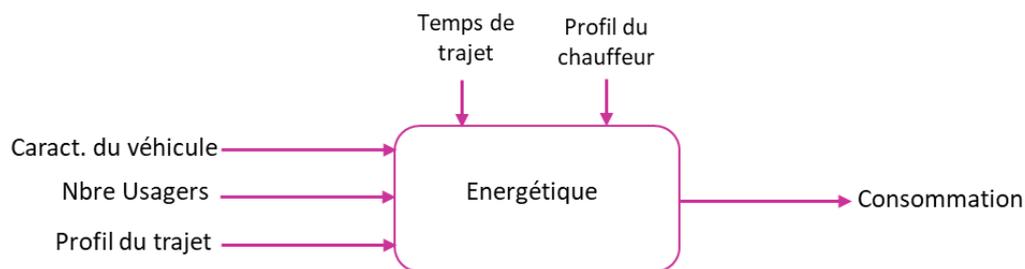


Figure 9 : Modèle Energétique de la mobilité

D'autres modèles sont également intéressants pour l'énergie comme celui qui donne les distances parcourus.

### 2.6.2 Environnementale

Ce modèle est en lien avec le précédent car la pollution dépend de la consommation du bus, mais également, de la distance parcourue. De plus, en ayant au préalable une connaissance du nombre d'individu à déplacer, le bus peut être adapté voire même supprimer ou proposé à la demande.

### 2.6.3 Economique

Ce modèle est important pour résoudre les problèmes liés aux précédents modèles cités. En effet, pour diminuer l'impact des transports sur l'environnement, le passage au bus électrique (voir chapitre 5) peut être une solution mais le coût lié à l'installation d'infrastructure, dont a besoin ce type de bus, reste un problème majeur.

### 2.6.4 Social

Ce modèle sert pour le placement des arrêts à des endroits stratégiques pour motiver les individus à opter pour les transports en commun. Il est également utile pour adapter le parcours du bus pour servir un maximum les usagers tout en roulant moins.

## 3. Fonctionnement de la simulation

### 3.1 Listes mémoire

La librairie<sup>4</sup> utilisée (ASQList.cpp) permet de gérer des listes en mémoire, dont les entrées (item) peuvent être soit de longueur variable soit de longueur fixe. Le programme développé est basé sur l'utilisation des fonctionnalités appliquées à des listes de longueur fixe, résumées dans le tableau suivant :

Nom de la fonction	Description
AddItemToFixedList ()	Ajouter un élément à la liste
GetItemFromFixedList ()	Chercher un élément dans une liste
GetItemCount ()	Donner le nombre d'élément d'une liste
CopyFixedListToFixedList ()	Copier une liste entière (même vide)
DeleteItemFromFixedList ()	Supprimer un élément d'une liste
FreeAndNullList ()	Effacer toute la liste de la mémoire

Fonction 1 : Fonctions utilisées pour les listes mémoires

De nouvelles fonctionnalités ont été créées sur la base de la librairie (ASQList.cpp) pour compléter celles qui ont été fournies. Le tableau suivant montre quelques-unes des nombreuses fonctions programmées pour travailler avec les listes :

Nom de la fonction	Description
GetStructId ()	Donner une ligne complète d'une liste
GetPositionInList ()	Donner la position d'un élément dans une liste
...	...

Fonction 2 : Fonctions complémentaires créées pour les listes mémoires

<sup>4</sup>Cette librairie a été fournie par mon professeur Gilbert Morand.

### 3.2 Evènement et Timer

Un évènement est un changement d'état produit par la validation d'une ou plusieurs conditions.

Un timer est une minuterie qui est lancée lorsqu'un évènement de déplacement de bus ou d'individus est appelé.

Les différents états et conditions sont représentés dans la machine d'état suivante :

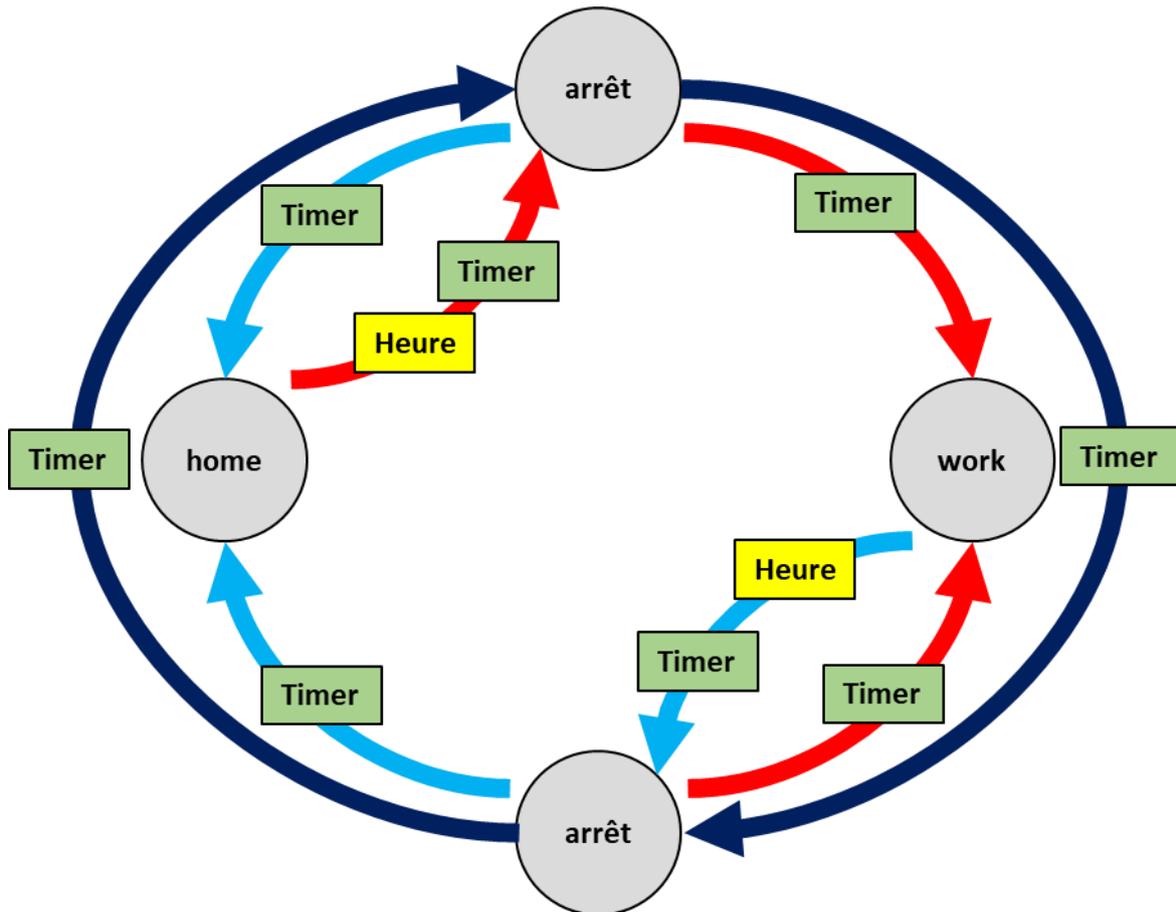


Figure 10 : Machine d'état

À la fin d'un timer ou si l'heure actuelle de la simulation est supérieure à l'heure de la condition, l'état actuel de la machine change. Lors d'un changement d'état, un évènement est déclenché.

Le schéma suivant résume le fonctionnement de la simulation en donnant la liste des évènements avec leur condition d'activation et leurs actions.

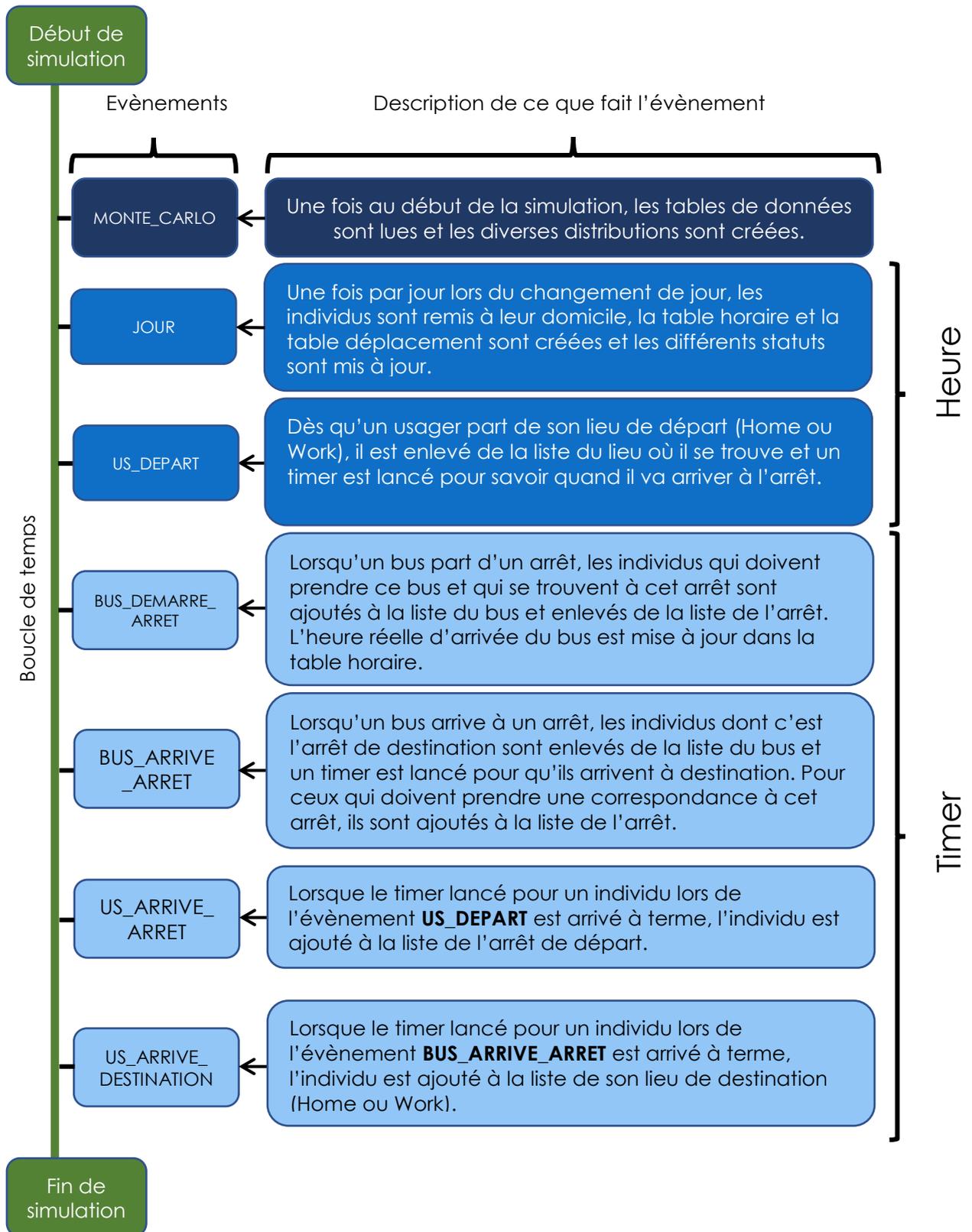


Figure 11 : Logique de simulation

### 3.3 Résumé de la simulation

Le programme simule une période temporelle définie par l'utilisateur en faisant « avancer le temps » seconde après seconde. La partie de l'interface utilisateur pour définir cette période est montrée ci-dessous :

Conditions temporelles de simulation	
Date et heure de départ	Date et heure de fin
2018/05/08 05:00:00	2018/05/08 23:59:00

Figure 12 : Définition de la période de simulation

Une fois la période temporelle de simulation choisit, la simulation peut commencer. Tout au long de la simulation les individus seront ajoutés à des listes mémoires ou supprimés de ces dernières lors des événements. Chaque lieu (maisons, arrêts, travaux, ...) et également chaque bus auront une liste mémoire rattachée. Dès qu'une personne se trouve dans un de ces endroits, il sera ajouté à la liste de l'endroit en question. De même lorsqu'un individu quitte un endroit, il sera supprimé de la liste qui y est rattachée. Lorsqu'un individu est en train de se rendre à un arrêt depuis un lieu ou depuis un arrêt à un lieu, il ne se trouvera dans aucune liste.

Voici un exemple de déplacements d'un individu (le 0 ici) depuis son domicile jusqu'à son lieu de travail.

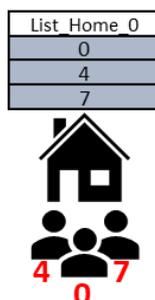


Figure 13 : L'individu 0 se trouve à son domicile (départ)

Sur la Figure 13 l'individu **0** se trouve chez lui en compagnie de soit sa famille s'il habite dans une maison individuelle ou soit des voisins s'il habite dans un immeuble.

Toutes les personnes qui habitent à cet endroit sont mises dans la liste **Home\_0** par l'évènement **JOUR**. Pendant la simulation, dès que l'évènement **US\_ARRIVE\_DESTINATION** est déclenché et que la destination d'individu est son domicile, il sera aussi ajouté à la liste de son domicile.

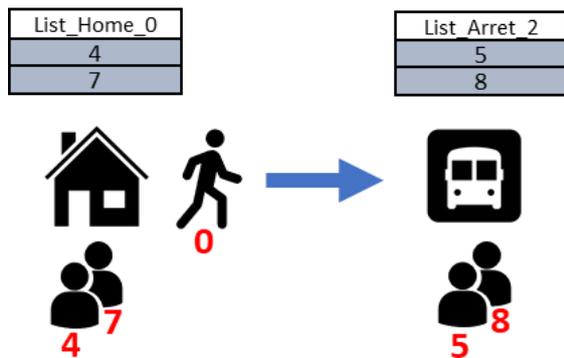


Figure 14 : l'individu 0 se rend à l'arrêt 2

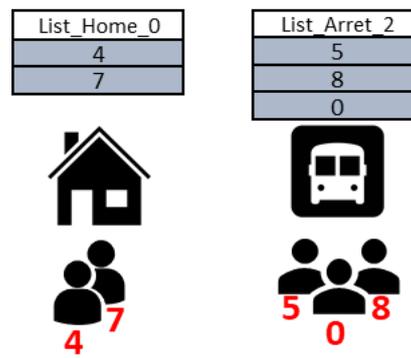


Figure 15 : l'individu 0 attend le bus

Lorsque l'évènement **US\_DEPART** est déclenché, l'individu **0** est supprimé de la liste **Home\_0** et un timer d'une durée de **TempsDep\_A** (Tableau 16) est lancé.

Lorsque le timer arrive à terme, l'évènement **US\_ARRIVE\_ARRET** est déclenché. Cela indique que l'individu est arrivé à l'arrêt et il est ajouté à la liste **Arrêt\_2** dans laquelle se trouvent d'autres usagers qui attendaient déjà un bus. (Figure 15)

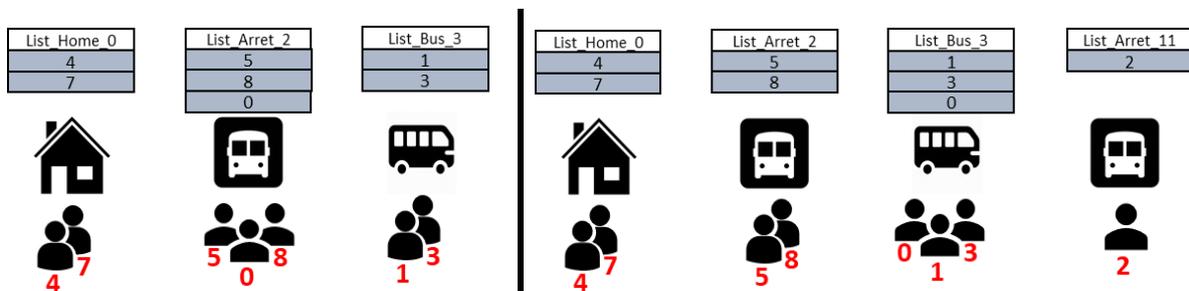


Figure 16 : Les bus passent par l'arrêt 2

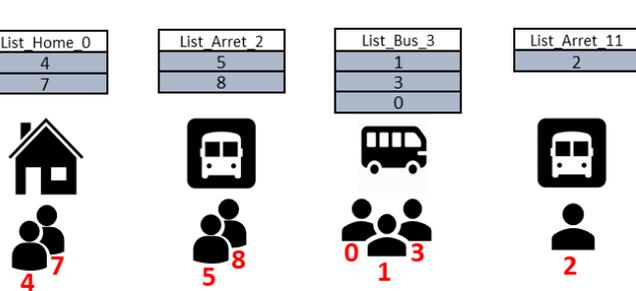


Figure 17 : L'individu 0 se trouve dans le bus 3

Plusieurs bus passent par l'arrêt 2 où l'individu **0** attend le bus en compagnie d'autres usagers. S'il s'agit du bon bus, lors de l'évènement **BUS\_DEMARRE\_ARRET**, l'individu **0** est ajouté à la liste **bus\_3** et il est retiré de la liste **Arret\_2**. (Figure 17)

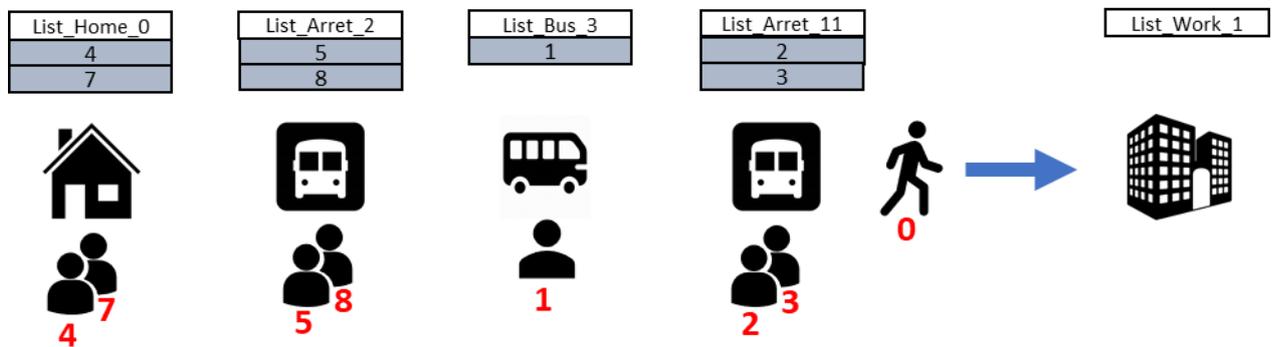


Figure 18 : l'individu 0 est arrivé à son arrêt de destination

Ensuite, le bus suivra sa ligne et s'arrêtera à chaque arrêt que contient son parcours lorsque le timer du trajet arrive à terme. Quand l'évènement **BUS\_ARRIVE\_ARRET** est déclenché et qu'il s'agit de l'arrêt de destination ou de correspondance d'un individu, il est enlevé de la **liste Bus\_3**.

Dans le cas d'une correspondance, comme ici pour l'individu 3 il est ajouté à la liste **Arret\_11** et il va recommencer les étapes depuis la Figure 15. L'autre possibilité est qu'il s'agit de l'arrêt de destination comme pour l'individu 0. Dans ce cas, un timer d'une durée de **TempsDest\_A** (Tableau 16) est lancé. (Figure 18)

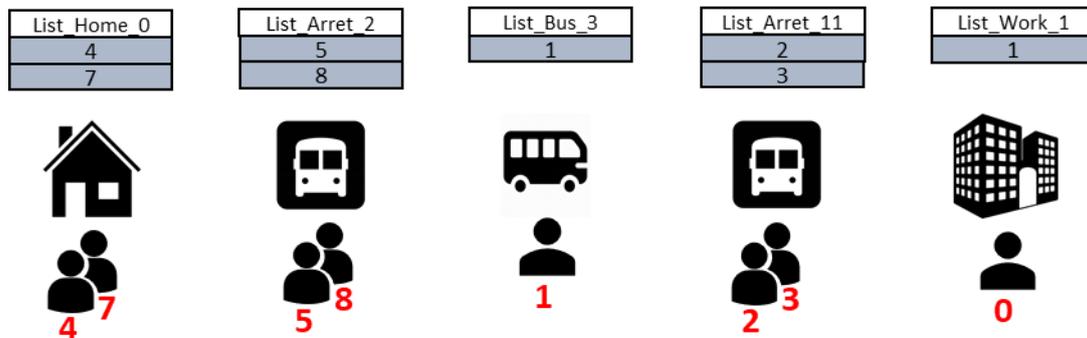


Figure 19 : l'individu 0 est arrivé à son lieu de travail (Destination)

Lorsque le timer arrive à terme, l'évènement **US\_ARRIVE\_DESTINATION** est déclenché. L'individu 0 sera ajouté à la liste **Work\_1** et le déplacement de l'individu est terminé. ( Figure 19)

### 3.4 Les tables de données à fournir en entrée de la simulation

Les paramètres à fournir pour le simulateur se trouvent dans différentes tables. Certaines de ces tables ont des liens entre elles. Deux tables, à partir de celles qui sont fournis sont générées par le logiciel.

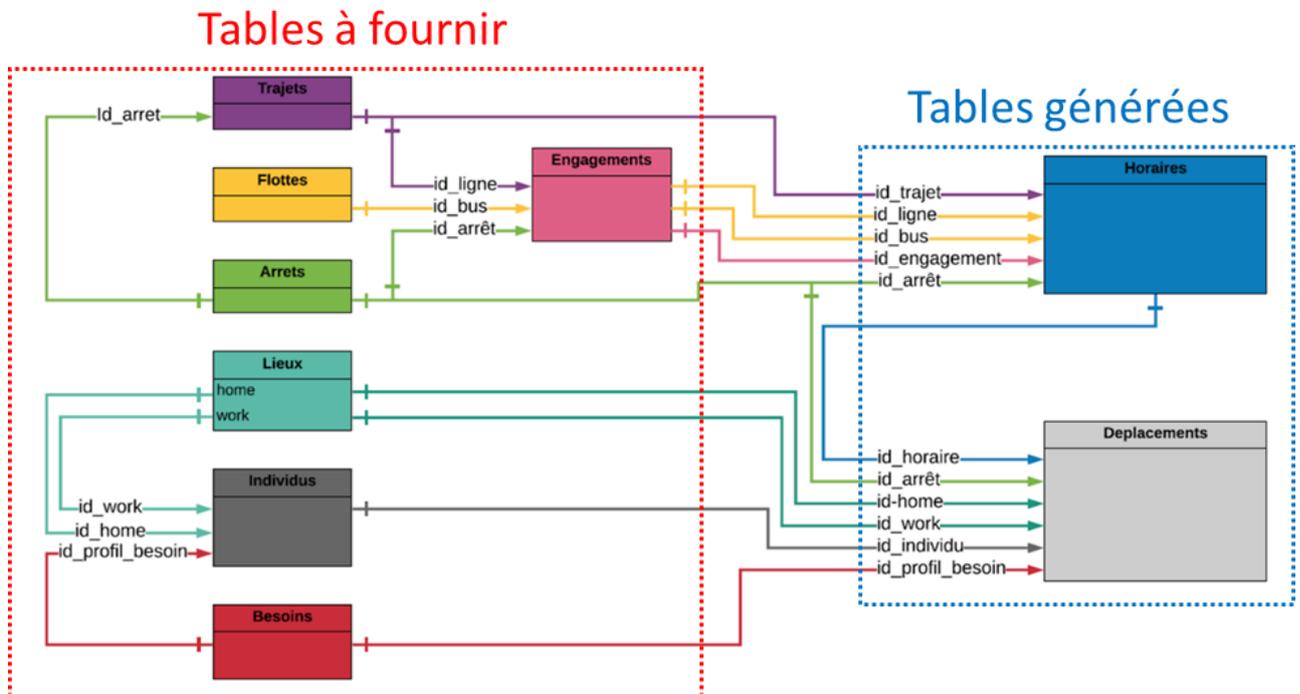


Figure 20 : Schéma relationnel entre les tables

Chaque table contient des informations précises. Le format et l'ordre dans lequel les tables sont fabriquées doivent être respectés. Dans le cas contraire, la simulation sera fautive ou ne fonctionnera pas. Pour éviter cela, chaque table va être décrite avec un exemple et une explication sera donnée pour comprendre les éléments à mettre dans les tables.

**Certains ids reviennent dans plusieurs tables. Lorsque c'est le cas, il suffit de regarder le nom qui suit l'id pour savoir à quelle table l'id fait référence.**

#### 3.4.1 Table Lieu\_Home

Cette table contient les informations sur les lieux de domicile où vivent les individus.

Home					
id_home	code_postal	ville	rue	latitude	longitude
0	1964	Conthey	Route des Peupliers 7a	46.220173	7.305238

Tableau 2 : Table Lieu\_Home

- **id\_home** : Un numéro id pour chaque immeuble ou maison est défini arbitrairement et ne doit pas se retrouver deux fois dans la même table.

Pour remplir les autres colonnes du tableau, l'exemple du Tableau 2 est explicite.

### 3.4.2 Table Lieu\_Work

Cette table contient les informations sur les lieux où les personnes peuvent se rendre comme des entreprises, des écoles, magasins, etc...

Work						
id_work	nom_lieu	code_postal	ville	rue	latitude	longitude
0	Magasin	1950	Sion	Rue de Pre-Fleuri 8B	46.229564	7.357019
1	HES SO Sion	1950	Sion	Rue du Rawil 47	46.241242	7.360131

Tableau 3 : Table Lieu\_Work

- **id\_work** : Un numéro id pour chaque lieu est défini arbitrairement et ne doit pas se retrouver deux fois dans la même table.

Pour remplir les autres colonnes du tableau, l'exemple du Tableau 3 est explicite.

### 3.4.3 Table flotte

Cette table contient les informations concernant la flotte de véhicule disponible.

Flotte						
id_bus	description	type	immatriculation	places_assises	places_debouts	consommation
0	Bus Diesel	Thermique	VS 12767	44	39	25
1	Bus Hybride	Hybride	VS 22767	44	39	15
2	Bus électrique 12m	Électrique	VS 32767	32	49	14,5

Tableau 4 : Table Flotte

- **id\_bus** : Un numéro id pour chaque bus est défini arbitrairement et ne doit pas se retrouver deux fois dans la même table.
- **type** : le type de bus est important pour calculer la consommation du bus
- **consommation** : La consommation moyenne sur 100 km de chaque véhicule. Pour les véhicules thermiques et hybride il faudra prendre en L/100 km et pour les bus électriques en kW/100 km.

Pour remplir les autres colonnes du tableau, l'exemple du Tableau 4 est explicite.

### 3.4.4 Table Arrêt

Cette Table contient les informations concernant les arrêts de bus.

Arrêts											
id_arret	nom_arret	code-postal	ville	rue	latitude	longitude	altitude	temps_arret	hub	recharge	abri
0	Depot	1964	Conthey	Route de la Morge	46.233945	7.310449	500	0	NON	NON	NON
1	Grands-Pres	1964	Conthey	Avenue de la Gare 44	46.225768	7.303530	492	30	OUI	NON	NON
2	Chateaneuf	1964	Conthey	Avenue de la Gare 16	46.219614	7.303627	486	30	OUI	NON	NON

Tableau 5 : Table Arrêts

- **id\_arret** : Un numéro id pour chaque arrêt est défini arbitrairement et ne doit pas se retrouver deux fois dans la même table. L'**id\_arret** est utilisé dans plusieurs table et grâce à cet id, toutes les informations concernant l'arrêt en question sont accessibles.
- **temps\_arret** : Pour l'instant le bus s'arrête 30 secondes à chaque arrêt. Par la suite, le temps d'arrêt du bus devra dépendre du nombre de personnes qui attendent et du temps de recharge si l'arrêt offre cette possibilité.

- **Hub** : **Oui** si l'arrêt permet un changement de ligne comme à la gare ou un arrêt qui va dans les deux directions. Ce type d'arrêts portent le même nom et se trouvent l'un en face de l'autre. **Non** si l'arrêt ne permet d'aller que dans une direction.
- **Recharge** : **Oui** si un bus électrique peut recharger ses batteries à cet arrêt. **Non** si la recharge est impossible.
- **Abri** : **Oui** si l'arrêt possède un abri pour les personnes. **Non** s'il n'en a pas.

Pour remplir les autres colonnes du tableau, l'exemple du Tableau 5 est explicite.

### 3.4.5 Table Trajets

Cette table contient les informations concernant les trajets que fait le bus. Un trajet est le parcours que le bus fait d'un premier arrêt à un second arrêt.

Trajets								
id_trajet	id_ligne	id_arret_depart	id_arret_arrivee	distance_reelle	temps_parcours_moyen	ecart_type_temps	vitesse_moyenne	id_profDepl
0	0	0	1	1500	600	40	40	1
1	1	1	2	681	150	10	20	1
2	1	2	3	1110	90	20	50	0
9	1	10	11	368	180	30	30	1
10	2	11	12	985	300	30	50	1
16	3	11	13	1500	480	50	40	1
17	3	13	14	700	40	10	50	0

Tableau 6 : Table Trajets

- **id\_trajet** : Un numéro id pour chaque trajet est défini arbitrairement et ne doit pas se retrouver deux fois dans la même table.
- **id\_ligne** : Un numéro id est défini pour une ligne qui se retrouve sur tous les trajets qui se trouvent sur cette ligne. A noter qu'une ligne complète n'as pas besoin d'être inscrite à la suite dans la table.
- **id\_arret\_depart** : C'est l'arrêt depuis lequel le bus part lors d'un trajet. (Tableau 5)
- **id\_arret\_arrivee** : C'est l'arrêt où le bus s'arrête lors d'un trajet. (Tableau 5)
- **distance\_reelle** : C'est la distance donnée par Google Maps lorsque les coordonnées GPS des deux arrêts sont donnée comme itinéraire.
- **temps\_parcours\_moyen** : C'est le temps que fait le bus pour parcourir la **distance\_reelle**. (Également donné par Google Maps)
- **ecart\_type\_temps** : nombre de secondes qui sera multiplié par un chiffre tiré au sort entre -1 et 1 dans le but varier le temps de parcours de quelques secondes. Ce chiffre est actuellement pris arbitrairement.
- **id\_profDepl** : C'est l'id du profil de déplacement. *A noter que pour le moment elle n'est pas utilisée et c'est l'ecart\_type\_temps qui fait varier le temps que fait le bus pour se rendre de son arrêt de départ à son arrêt de destination.*

Cette table sera utile pour la table Engagement ainsi que pour la table Horaire car elle comporte toutes les indications sur les trajets qu'empruntent les bus. (Les lignes surlignées en jaunes serviront pour expliquer d'autres fonctions plus tard.)

### 3.4.6 Table Individus

Cette table contient les informations des individus.

Individus						
id_individu	annee_naissanc	permis	vehicule	id_profil_besoin	id_home	id_work
0	1992	OUI	OUI	0	0	1
1	1981	NON	NON	1	1	2

Tableau 7 : Table Individus

- **id\_individu** : Un numéro id unique est attribué à chaque individu. Ce chiffre est attribué arbitrairement.
- **id\_profil\_besoin** : Ce numéro indique le profil de l'individu. Le numéro représente un profil qui est défini dans la table profil besoin. (Tableau 8)

Pour remplir les autres colonnes du tableau, l'exemple du Tableau 7 est explicite.

### 3.4.7 Table Besoins

Cette table contient les besoins en mobilité pour les différents profils des utilisateurs. De plus, elle contient des informations comme les heures d'arrivée qui varient selon le profil ainsi que les chances de prendre le bus.

Besoin					
id_profil_besoin	id_besoin	heure_arrivee	pUtilisationBus	vitesse_marche	ecart_type
0	0	08:00	1,00	3,20	0,3
0	1	12:00	0,40	3,50	0,3
0	2	13:00	0,40	3,10	0,3
0	3	17:00	0,80	3,20	0,3
1	0	07:00	1,00	3,70	0,3
1	1	17:00	1,00	3,10	0,3

Tableau 8 : Table Besoin

- **id\_profil\_besoin** : Un numéro id indique le profil de la personne, son nombre de déplacements quotidien ainsi que l'heure à laquelle il doit arriver une destination. Ici le **0** est utilisé pour tous les étudiants qui rentrent à midi et le **1** pour les employées qui prennent le bus que le matin et le soir. Le Tableau 9 montre quelques exemples de profils qui existent. A noter que la liste des profils n'est pas exhaustive.
- **id\_besoin** : Il indique le numéro du déplacement. Pour chaque profil ce numéro commence à 0.
- **heure\_arrivee** : Il s'agit de l'heure à laquelle l'individu commence le travail ou les cours. C'est une indication pour savoir quel bus l'individu prendra. Un tirage de la ponctualité sera effectué pour avoir l'heure d'arrivée réelle de chaque individu. (Voir Figure 4)
- **pUtilisationBus** : Il s'agit de la probabilité qu'a un individu de prendre le bus et celle-ci doit être plus petit ou égale à 1. Cette valeur est comparée à un autre chiffre tiré au sort et qui se situe entre 0 et 1.

- **vitesse\_marche** : Il s'agit de la vitesse de marche moyenne de l'individu en km/h. Cette information est utilisée pour avoir approximativement le temps que fait l'individu pour se rendre à pied aux arrêts de bus ou au lieu de destination depuis les arrêts.
- **ecart\_type** : il s'agit d'une vitesse en km/h qui sera multipliée par un chiffre tiré au sort entre -1 et 1 pour faire varier la **vitesse\_marche** de l'individu lors de chaque besoin. Ce chiffre est actuellement choisi arbitrairement.

Besoin			
id_profil_ besoin	Profil	Nombre de déplacement	heure_arrivee
0	Etudiant	4x	08:00
1	Employé	2x	07:00
2	Etudiant	2x	08:00
3	Employé	4x	06:30
4	Écolier	2x	09:00
5	Ecolier	4x	09:00
6	Employé	2x	libre
7	Employé	4x	libre
8	Retraité	2 ou 0	variable
9	Chômeur	2 ou 0	variable

Tableau 9 : Exemples de profil

### 3.4.8 Table Engagements

Cette table contient les informations liées à la mise en place d'un horaire de bus. Elle donne l'id du bus qui est engagé, l'heure de départ, l'arrêt d'où le bus commence son parcours ainsi que la ligne qu'il va suivre.

Engagements				
id_engagement	id_bus	id_ligne	id_arret_depart	heure_depart
0	0	0	0	06:45
1	1	0	0	07:00
2	0	1	1	07:00
3	1	1	1	07:15
4	2	2	11	07:25
5	3	3	11	07:36

Tableau 10 : Table Engagements

- **id\_engagement** : Un numéro id unique est attribué à chaque engagement. Ce chiffre est attribué arbitrairement.
- **id\_arret\_depart** : Il s'agit de l'arrêt du lequel le bus commence son parcours. En début de journée, lorsque le bus ne commence pas son parcours à la gare, il part d'un dépôt qui se trouve ici à l'arrêt **0**. (Tableau 5)
- **heure\_depart** : Il s'agit de l'heure à laquelle le bus commence son parcours.

Cette table peut être remplie en suivant un horaire réel de bus. Cependant, il faut que les heures de départ soient dans un ordre chronologique.

### 3.5 Les tables de données générées par le logiciel de simulation

Les tables précédemment expliquées sont à fournir par l'utilisateur. Les deux prochaines tables sont générées par le logiciel à partir des tables fournies comme le montre la Figure 20. Elles sont créées à l'aide de plusieurs fonctions qui seront détaillées par la suite.

#### 3.5.1 Table Horaire

Cette table horaire est établie une fois par jour à 00h00 lorsque l'évènement JOUR est appelé (voir 3.2).

Horaire															
id_horaire	id_engagement	id_bus	id_ligne	id_trajet_ligne	id_arret_depart	id_arret_arrivee	Heure_planifie_depart	Heure_planifie_arrivee	Heure_Reel_depart	Heure_Reel_arrivee	Places_Assises	Places_Debords	Places_Totals	Places_Libres	Statut
0	0	0	0	0	0	1	06:45:00	06:55:00	06:45:00	06:55:28	20	10	30	30	3
1	1	1	0	0	0	1	07:00:00	07:10:00	07:00:00	07:09:19	30	15	45	45	2
2	2	0	1	1	1	2	07:15:00	07:17:30	07:15:00		20	10	30	30	10
3	2	0	1	2	2	3	07:18:00	07:19:30			20	10	30	30	0
11	3	1	1	1	1	2	07:25:00	07:27:30	07:25:00		30	15	45	45	10
12	3	1	1	2	2	3	07:28:00	07:29:30			30	15	45	45	0
20	4	2	2	10	11	12	07:43:00	07:48:00	07:43:00		30	15	45	45	10
21	4	2	2	11	12	7	07:48:30	07:50:00			30	15	45	45	0
26	5	3	3	16	11	13	07:36:00	07:44:00	07:36:00		20	10	30	30	10
27	5	3	3	17	13	14	07:44:30	07:45:10			20	10	30	30	0

Tableau 11 : Table horaire en cours de création

- **id\_horaire** : Un numéro id unique est attribué à chaque déplacement de la table horaire. Ce chiffre est attribué arbitrairement.

Les informations suivantes sont copiées à partir des précédentes tables :

- **id\_engagement** : il s'agit de l'id de l'engagement qui vient de la table Engagements (Tableau 10). Il reste le même jusqu'à ce que le bus arrive au bout de la ligne.
- **id\_trajet\_ligne** : Il s'agit de l'id du trajet que le bus doit utiliser. (Tableau 6)

Les informations suivantes sont calculées à partir de fonctions :

- **heure\_planifie\_depart** et **heure\_planifie\_arrivee** :

Lorsqu'il s'agit du départ du bus au début d'une ligne l'**heure\_planifie\_depart** est copié depuis la table Engagements (Tableau 10). Ensuite pour trouver l'**heure\_planifie\_arrivee** la fonction suivante est utilisée :

```
char *GetTempsArrivee(char *heureMinuteStrPlanDepart,int dureeTrajet);
```

Fonction 3 : GetTempsArrivee

Cette fonction va prendre en argument l'**heure\_planifie\_depart** et la durée du trajet qui se trouve dans le Tableau 6 pour le trajet en question et va donner l'**heure\_planifie\_arrivee**.

Ensuite, lorsqu'il ne s'agit pas du départ d'une ligne, l'**heure\_planifie\_depart** sera l'**heure\_planifie\_arrivee** à laquelle il sera rajouté le temps d'arrêt qui se trouve dans le Tableau 5 pour l'arrêt en question. Les heures planifiées de départ et d'arrivée suivants seront remplis automatiquement car il s'agira de l'horaire du bus qui servira de planning pour les usagers.

- **heure\_Reelle\_depart** et **heure\_Reelle\_arrivee** :

Pour l'**heure\_Reelle\_depart**, lorsqu'il s'agit du départ du bus au début d'une ligne, l'**heure\_planifie\_depart** est recopiée. C'est le cas des horaires surlignés en vert dans le Tableau 11. Une fois cette case remplie, un timer (minuterie) est lancé pour savoir quand le bus part.

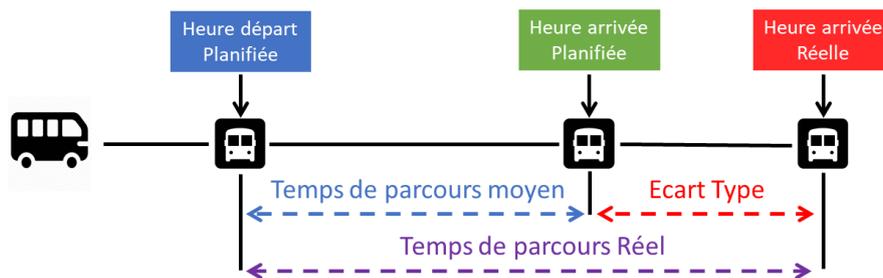


Figure 21 : Heure arrivée réelle du bus en retard

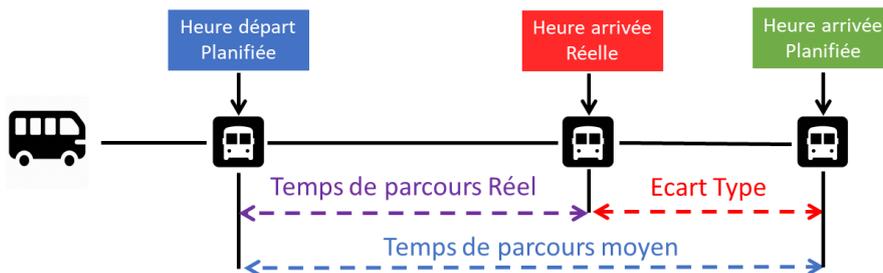


Figure 22 : Heure arrivée réelle du bus en avance

Une fois que le temps du timer est écoulé, l'évènement **BUS\_DEMARRE\_ARRET** est appelé. Cet évènement va rajouter l'**heure\_Reelle\_arrivee**, en prenant l'**heure\_Reelle\_depart** et en lui rajoutant la durée du trajet avec l'écart type du trajet en question qui se trouve dans le Tableau 6. A noter que l'écart type est multiplié par un tirage de Gauss pour se rapprocher de la réalité en prenant en compte les aléas de circulation. De plus, un timer sera lancé pour savoir quand le bus va arriver à destination.

Lorsque le temps de ce nouveau timer est dépassé, l'évènement **BUS\_ARRIVE\_ARRET** est appelé. Cet évènement va rajouter l'**heure\_Reelle\_depart** pour l'horaire suivant. Pour cela, la fonction suivante est utilisée :

```
struct HORAIRE * GetNextiHo(char *listBin,int id_engagement,
                           int id_arret_arrivee);
```

Fonction 4 : GetNextiHo

Cette fonction va prendre en argument la table Horaire, l'**id\_engagement** et l'**id\_arret\_arrivee** de l'horaire en question.

Si la fonction retourne quelque chose, cela voudra dire que la ligne n'est pas encore terminée et que l'**heure\_Reelle\_depart** du bus suivant doit être rempli. Pour cela deux cas sont distingués :

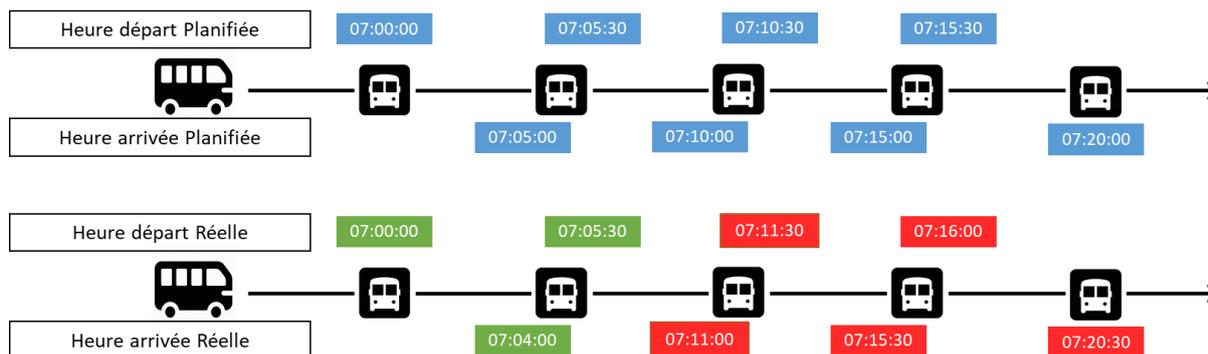


Figure 23 : Horaire planifié et réel

- **Cas 1 en vert** : Le bus arrive avant ou pile à l’heure\_planifie\_arrivee. Dans ce cas, l’heure\_Reelle\_depart sera égal à heure\_planifie\_depart.
  - **Cas 2 en rouge** : Le bus arrive après l’heure\_planifie\_arrivee. Dans ce cas, l’heure\_Reelle\_depart sera l’heure\_Reelle\_arrivee à laquelle sera additionné le temps d’arrêt.
- **Statut** : Il s’agit du statut du bus en temps réel.

N° du statut	Description
0	Le bus se trouve en début de ligne
1	Heure de départ réelle du bus connu
10	Timer lancé pour le départ
2	Heure d’arrivée réelle du bus connu
20	Timer lancé pour l’arrivée
3	Le bus est arrivé à destination

Tableau 12 : Statuts de l’horaire du bus

Voici un exemple d’une partie de la table horaire lorsque la simulation est terminée. Tous les statuts sont à 3, ce qui veut dire que le bus est bien arrivé à destination pour l’horaire.

Horaire															
id_horaire	id_engagement	id_bus	id_ligne	id_trajet_ligne	id_arret_depart	id_arret_arrivee	Heure_planifie_depart	Heure_planifie_arrivee	Heure_Reel_depart	Heure_Reel_arrivee	Places_Assises	Places_Debouts	Places_Totals	Places_Libres	Statut
0	0	0	0	0	0	1	06:45:00	06:55:00	06:45:00	06:55:28	20	10	30	30	3
1	1	1	0	0	0	1	07:00:00	07:10:00	07:00:00	07:09:19	30	15	45	45	3
2	1	0	1	1	1	2	07:15:00	07:17:30	07:15:00	07:17:37	20	10	30	29	3
3	1	0	1	2	2	3	07:18:00	07:19:30	07:18:07	07:19:33	20	10	30	27	3
4	1	0	1	3	3	4	07:20:00	07:20:40	07:20:03	07:20:43	20	10	30	27	3
5	1	0	1	4	4	5	07:21:10	07:21:50	07:21:13	07:21:45	20	10	30	27	3
6	1	0	1	5	5	6	07:22:20	07:23:00	07:22:20	07:23:10	20	10	30	27	3
7	1	0	1	6	6	8	07:23:30	07:26:00	07:23:40	07:26:18	20	10	30	27	3
8	1	0	1	7	8	9	07:26:30	07:28:00	07:26:48	07:28:28	20	10	30	27	3
9	1	0	1	8	9	10	07:28:30	07:30:00	07:28:58	07:30:03	20	10	30	27	3
10	1	0	1	9	10	11	07:30:30	07:33:30	07:30:33	07:33:00	20	10	30	27	3
11	2	1	1	1	1	2	07:25:00	07:27:30	07:25:00	07:27:26	30	15	45	42	3
12	2	1	1	2	2	3	07:28:00	07:29:30	07:28:00	07:29:15	30	15	45	28	3
13	2	1	1	3	3	4	07:30:00	07:30:40	07:30:00	07:30:35	30	15	45	28	3
14	2	1	1	4	4	5	07:31:10	07:31:50	07:31:10	07:31:53	30	15	45	28	3
15	2	1	1	5	5	6	07:32:20	07:33:00	07:32:23	07:32:56	30	15	45	28	3
16	2	1	1	6	6	8	07:33:30	07:36:00	07:33:30	07:35:27	30	15	45	28	3
17	2	1	1	7	8	9	07:36:30	07:38:00	07:36:30	07:38:03	30	15	45	28	3
18	2	1	1	8	9	10	07:38:30	07:40:00	07:38:33	07:40:01	30	15	45	33	3
19	2	1	1	9	10	11	07:40:30	07:43:30	07:40:31	07:43:18	30	15	45	39	3
20	3	2	2	10	11	12	07:43:00	07:48:00	07:43:00	07:47:42	30	15	45	45	3
21	3	2	2	11	12	7	07:48:30	07:50:00	07:48:30	07:50:03	30	15	45	45	3
22	3	2	2	12	7	4	07:50:30	07:51:00	07:50:33	07:50:58	30	15	45	45	3
23	3	2	2	13	4	3	07:51:30	07:52:00	07:51:30	07:51:56	30	15	45	45	3
24	3	2	2	14	3	2	07:52:30	07:54:00	07:52:30	07:53:45	30	15	45	45	3
25	3	2	2	15	2	1	07:54:30	07:56:00	07:54:30	07:55:47	30	15	45	45	3
26	4	3	3	16	11	13	07:36:00	07:44:00	07:36:00	07:44:07	20	10	30	28	3
27	4	3	3	17	13	14	07:44:30	07:45:10	07:44:37	07:45:28	20	10	30	28	3
28	4	3	3	18	14	15	07:45:40	07:46:20	07:45:58	07:46:42	20	10	30	30	3
29	4	3	3	19	15	16	07:46:50	07:48:20	07:47:12	07:48:45	20	10	30	30	3

Tableau 13 : Partie de la table Horaire finale

Evidemment le tableau ne sera pas sous ce format-là. (Les lignes surlignées en jaunes serviront à expliquer d’autres fonctions plus tard.)

### 3.5.2 Table Déplacement

Cette table déplacement est établie une fois par jour à 00h00 lorsque l’évènement JOUR est appelé, puis le statut sera mis à jour tout au long de la simulation.

Pour des raisons de lisibilité, la table de déplacement a été découpée en plusieurs parties.

Déplacement									
id_deplacement	id_individu	id_profil_besoin	id_lieu_depart	id_lieu_destination	Latitude_Depart	Longitude_Depart	Latitude_Destination	Longitude_Destination	Besoin_Bus
0	0	0	0	1	46.220173	7.305238	46.241242	7.360131	1
1	0	0	1	0	46.241242	7.360131	46.220173	7.305238	1
2	0	0	0	1	46.220173	7.305238	46.241242	7.360131	1
3	0	0	1	0	46.241242	7.360131	46.220173	7.305238	1
4	1	1	1	2	46.222200	7.299373	46.233988	7.355464	1
5	1	1	2	1	46.233988	7.355464	46.222200	7.299373	1

Tableau 14 : Table Déplacements Partie 1

- **id\_deplacement** : Un numéro id unique est attribué à chaque déplacement d'un individu. Ce chiffre est attribué arbitrairement.
- **Besoin\_Bus** : Pour définir si un individu va prendre le bus ou pas, la fonction suivante est utilisée :

```
int ReturnChance(float probability);
```

Fonction 5 : ReturnChance

Cette fonction va tirer au sort un nombre entre 0 et 1 et va comparer le nombre tiré avec celui qui est en argument. Ce dernier est la **pUtilisationBus** qui se trouve dans la table Besoin. (Tableau 8) Si le chiffre tiré est plus petit, alors l'individu va prendre le bus et **Besoin\_Bus sera mis à 1**. Dans le cas contraire, la ligne dans la table déplacement (Tableau 14) pour l'individu est supprimée et l'individu suivant sera traité.

Le reste des informations est recopié en fonction des lieux de départ et de destination des individus.

Déplacement							
	Heure_Arrivee_Souhaitee	id_arret_Depart	id_arret_Destination	Latitude_Arret_Depart	Longitude_Arret_Depart	Latitude_Arret_Destination	Longitude_Arret_Destination
08:00:00	2	14	46.219612	7.303627	46.241001	7.360544	
12:00:00	14	2	46.241001	7.360544	46.219612	7.303627	
13:00:00	2	14	46.219612	7.303627	46.241001	7.360544	
17:00:00	14	2	46.241001	7.360544	46.219612	7.303627	
07:00:00	2	9	46.219612	7.303627	46.231758	7.356913	
17:00:00	9	2	46.231758	7.356913	46.219612	7.303627	

Tableau 15 : Table Déplacement Partie 2

- **Heure\_arrivee\_souhaite** : Il s'agit de l'heure à laquelle l'individu doit, au plus tard arriver à son lieu de destination. Il est copié depuis la table besoin. (Tableau 8). A noter que pour le soir, il serait probablement plus adapté de mettre l'heure de départ du lieu work à la place de mettre l'heure d'arrivée du lieu home.
- **id\_arret\_Depart et id\_arret\_destination** : Pour déterminer l'arrêt de départ et de destination le plus adapté pour chaque individu, la fonction suivante est utilisée :

```
int ChercheArretPlusProche(struct LISTS_MOBILITY *mLists,
double latitude, double longitude);
```

Fonction 6 : ChercheArretPlusProche

Pour l'instant, il a été décidé que l'arrêt le plus adapté est celui qui se trouve le plus proche de la position de l'individu et c'est ce qu'effectue la *Fonction 6* en prenant en argument la position de départ et de destination de ce dernier, il donne l'arrêt le plus proche pour les deux cas. A noter que c'est bien l'arrêt de destination final qui est indiqué, même dans le cas d'un changement de bus pour une correspondance.

Le reste des informations est recopié en fonction des arrêts de départ et de destination des individus.

Déplacement									
	DistanceDep_A	TempsDep_A	DistanceA_Dest	TempsA_Dest	Correspondance	Heure_Depart	Temps TotalDepl	DistTotalDepl	Activite_Actuelle
131.99	137.26	37.16	42.00	1	07:15:23	43	7512	6	
37.16	36.48	131.99	142.01	1	11:18:52	36	6934	6	
131.99	155.34	37.16	57.65	1	12:17:46	41	7512	6	
37.16	45.98	131.99	150.24	1	16:24:05	44	6934	6	
388.03	394.15	211.11	195.85	0	05:59:57	20	4707	6	
211.11	221.78	388.03	419.36	1	16:19:48	44	5339	6	

Tableau 16 : Table Déplacement Partie 3

- **DistanceDep\_A et DistanceA\_Destv** : Une fois que les arrêts les plus proches du lieu de départ et de destination sont trouvés, il faut trouver la distance qui les sépare. Pour cela plusieurs fonctions ont été utilisées. (Fonction 10 : ComputeDist)
- **TempsDep\_A et TempsDest\_A** : Après avoir trouvé la distance qui sépare les lieux et les arrêts, avec la vitesse de marche et l'écart type présent dans la table besoin (Tableau 8) le temps est déterminé en utilisant l'équation suivante :

$$temps [s] = \frac{distance [m]}{vitesse [\frac{m}{s}]}$$

Équation 2 : Equation de la vitesse

- **Correspondance** : Donne le nombre de correspondances. Si c'est un **0** cela veut dire que l'individu n'as pas besoin de changer de bus. Cette information est donnée par une fonction qui va trouver le chemin que l'individu doit parcourir pour optimiser son temps de voyage et ainsi donner le nombre de correspondances. Cette fonction est expliquée au chapitre (Voir Figure 35 3.6.3).
- **Heure\_depart** : Il s'agit de l'heure à laquelle l'individu doit parti de son lieu de départ.

Horaire															
id_horaire	id_engagement	id_bus	id_ligne	id_trajet_ligne	id_arret_depart	id_arret_arrivee	Heure_planifie_depart	Heure_planifie_arrivee	Heure_Reel_depart	Heure_Reel_arrivee	Places_Assises	Places_Debouts	Places_Totals	Places_Libres	Statut
26	5	3	3	16	11	13	07:36:00	07:44:00	07:36:00		20	10	30	30	0
3	2	0	1	2	2	3	07:18:00	07:19:30			20	10	30	30	1

Tableau 17 : Exemple de liste déplacement individu

En ayant le bon horaire, **Heure\_depart** est déterminée comme le montre la Figure 24. (L'image se trouve en plus grande à l'annexe 2.)

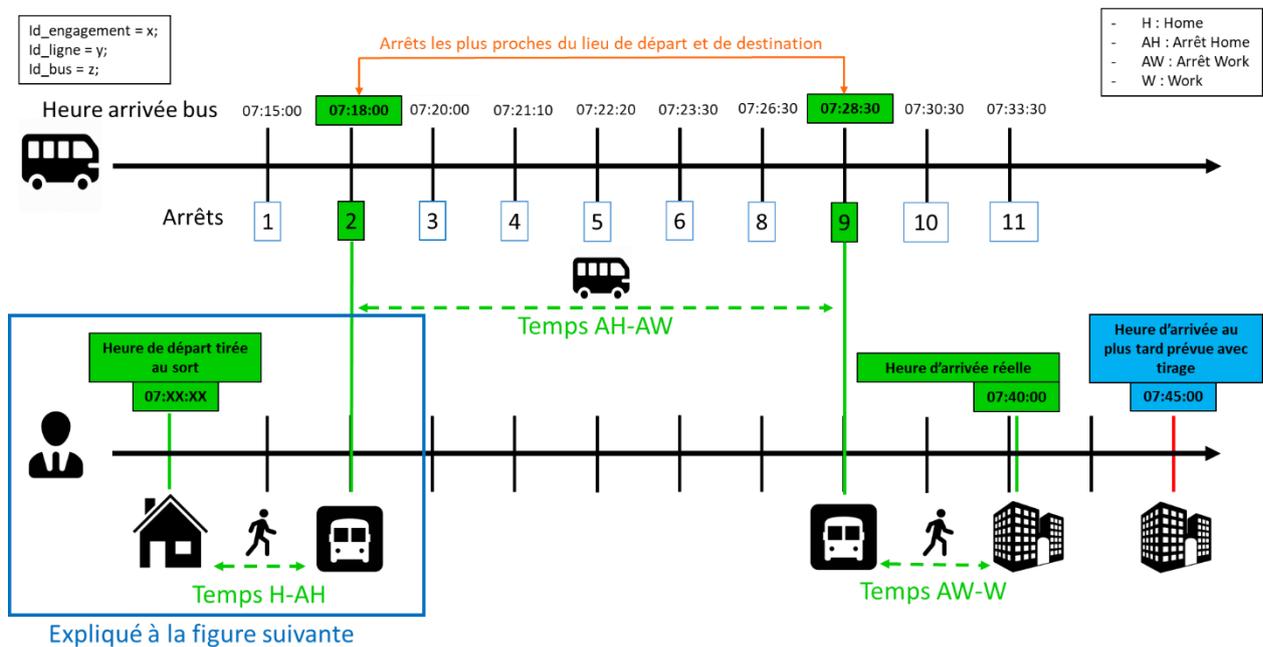


Figure 24 : Schéma déplacement avec 0 correspondance

Comme indiqué sur la Figure 24, l'arrêt de départ est le N° 2 et celui de destination est le N°9. Pour que l'individu arrive à destination à l'heure d'arrivée prévue avec le tirage (Figure 4) ou un peu avant, il doit prendre le bus qui est planifié à l'arrêt N°2 à 07h18. Pour qu'il se déplace jusqu'à cet arrêt aux alentours de l'heure d'arrivée du bus, un tirage sera effectué avec le **tempsDep\_A** ± le tirage depuis une fonction de répartition (Figure 25). Une fois que l'heure de départ a été déterminée, un timer est lancé comme le montre la figure suivante.

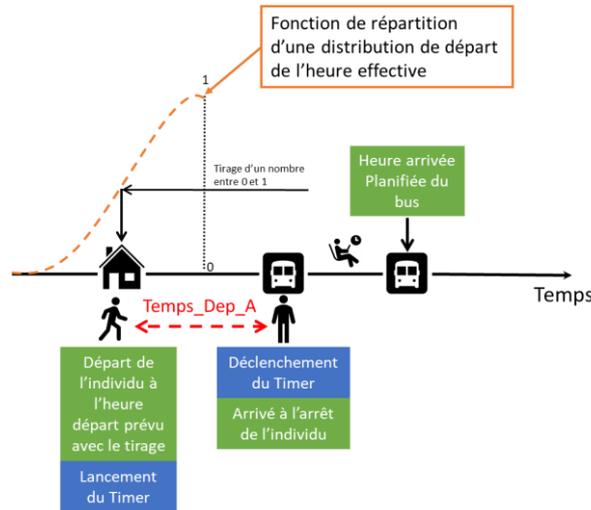


Figure 25 : Lancement du Timer pour le déplacement jusqu'à l'arrêt

Dans le cas d'un changement de bus, l'individu doit pouvoir prendre sa correspondance. Pour cela, un délai minimum de 3 minutes est accordé pour le changement de bus. La manière de déterminer l'heure de départ de l'individu ne change pas dans le cas où il y a une correspondance. (L'image est disponible en plus grande à l'annexe 3.)

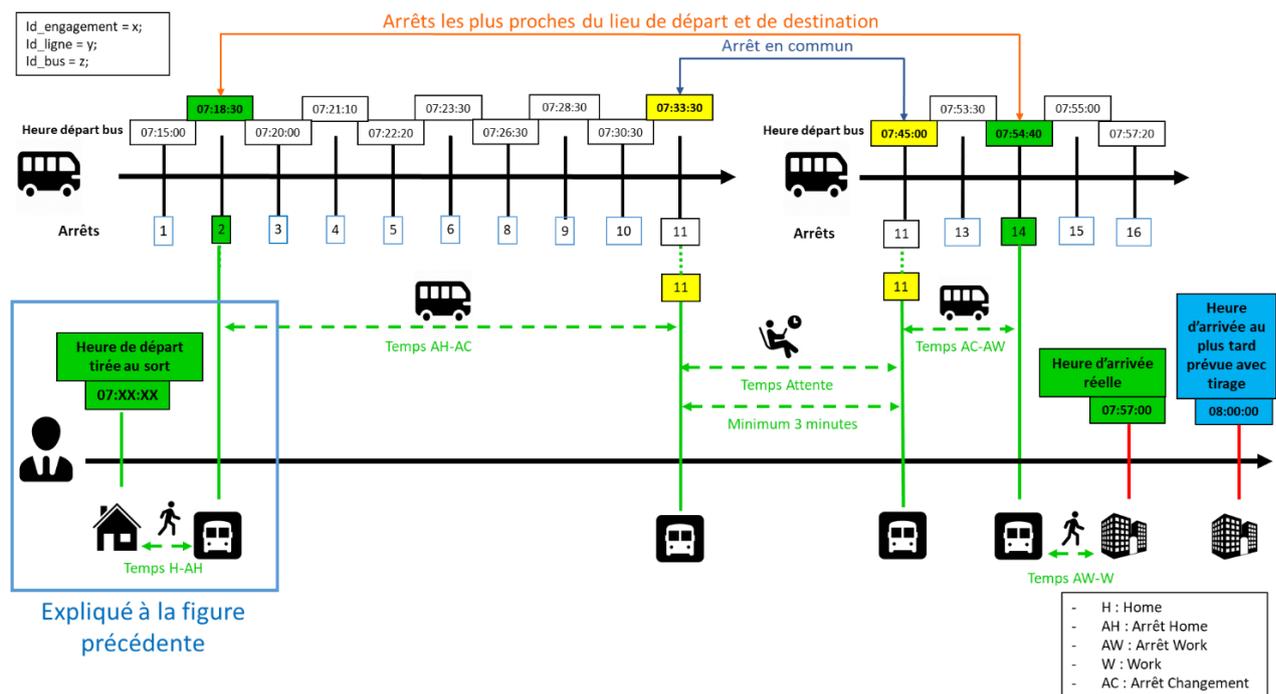


Figure 26 : Schéma déplacement avec 1 correspondance

- TempsTotalDepl** : Il s'agit du temps total que met un individu pour se rendre depuis son lieu de départ jusqu'à son lieu de destination. Dans ce temps est compris le **TempsDep\_A**, **TempsDest\_A**, le temps d'attente du bus, le temps d'attente de la correspondance dans le cas d'un changement ainsi que le temps de trajet en bus. C'est pour cela que dans le Tableau 16, pour l'individu N°1 le temps d'aller est de 20 minutes et celui du retour est de 44 minutes. En effet, pour le retour il doit prendre une correspondance. Ceci explique la grande différence de temps entre l'aller et le retour.

- **DistTotalDepl** : Il s'agit de la distance totale que parcourt un individu en bus.
- **Activite\_Actuelle** : Il s'agit de ce que fait l'individu en temps réel.

N° de l'activité	Description
0	Pas concerné
1	Doit bientôt aller prendre le bus
2	Va prendre le bus
3	Attend le bus
4	Se trouve dans le bus
5	Sort du bus
6	Est arrivé à destination

Tableau 18 : Activité de l'individu

Lorsque l'activité actuelle d'un individu n'est pas mise à la valeur 6, cela veut dire qu'il est bloqué quelque part. Pour savoir où il se trouve, il suffit de regarder l'activité actuelle qui est affichée. Si tous les usagers ont pu prendre le bus est arriver à destination pour chacun de leur déplacement à la fin de journée, l'activité actuelle de tous les déplacements doit être à 6.

### 3.6 Fonctions importantes programmées

Dans ce chapitre, toutes les fonctions importantes seront expliquées.

Une structure<sup>5</sup> en programmation permet de regrouper plusieurs variables de différents types dans une même entité. Ainsi il est possible de construire de nouveaux types plus complexes. Toutes les tables présentées jusqu'ici se trouvent dans une structure.

#### 3.6.1 Gestions du temps

Le temps peut être interprété sous plusieurs formes, comme par exemple en nombre d'années, d'heures, ou plus précisément, de secondes. Les calculs se feront en secondes mais pour une question de compréhension, certaines réponses doivent être données sous d'autres formes. Pour cela, la structure suivante contient les informations de temps sous toutes les formes courantes.

```
struct MON_TEMPS
{
    char dateHeureStr    [40];
    char dateStr        [20];
    char heureStr        [20];
    int jour;
    int mois;
    char moisStr         [20];
    int annee;
    int heure;
    int minute;
    int seconde;
    int iJourSemaine;
    char jourSemaineStr [20];
    long t70;
};
```

Figure 27 : Structure MON\_TEMPS

<sup>5</sup> [https://fr.wikibooks.org/wiki/Programmation\\_C%2B%2B/Les\\_structures](https://fr.wikibooks.org/wiki/Programmation_C%2B%2B/Les_structures)

Avec cette structure, la gestion du temps a besoin des fonctions suivantes :

La première fonction est la suivante :

```
int Determine_t70Horaire(char *heureMinuteStr, struct MON_TEMPMS *mon_t);
```

#### Fonction 7 : Determine\_t70Horaire

Cette fonction donne le nombre de secondes qu'il y a depuis le 1 janvier 1970 jusqu'au moment souhaité. Ce moment est celui qui est donné en argument de la fonction. Dans `char *heureMinuteStr` il faut mettre l'heure, et dans `MON_TEMPMS *mon_t` il faut mettre le temps actuel de la simulation pour avoir la date (année, mois, jour).

La seconde fonction fait l'inverse :

```
int From_t70_to_MON_TEMPMS(int t70, struct MON_TEMPMS *mon_t);
```

#### Fonction 8 : From\_t70\_to\_Mon\_Temps

En donnant en argument un nombre de secondes, la fonction retourne l'argument sous toutes les formes de la structure `MON_TEMPMS`.

La dernière fonction est utilisée dans la Fonction 8 :

```
int AssigneStrut_MON_TEMPMS(char *dateHeureStr, struct MON_TEMPMS *mon_t);
```

#### Fonction 9 : AssigneStruct\_MON\_TEMPMS

C'est elle qui sépare les différentes informations de temps pour les mettre sous la bonne forme et ainsi pouvoir les retourner sous la forme de la structure `MON_TEMPMS`. Tous les calculs et toutes les comparaisons se font avec le nombre de secondes qu'il y a entre deux événements.

### 3.6.2 Calcul de distance<sup>6</sup>

Les informations de position des arrêts et des bâtiments sont données en coordonnée GPS par la latitude et la longitude dans les différentes tables de donnée.

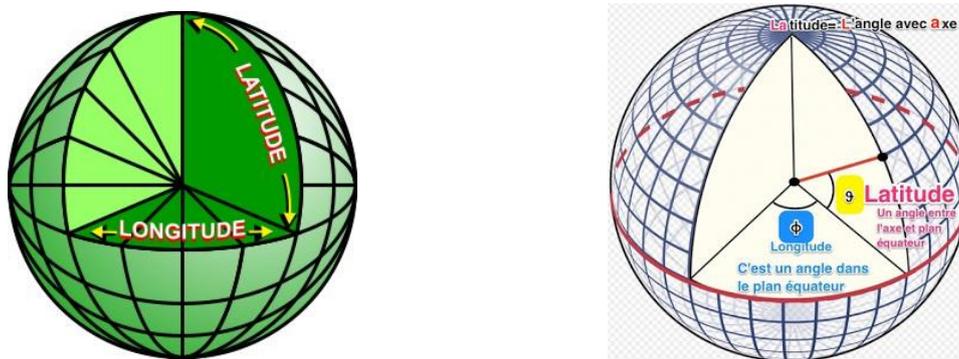


Figure 28 : Définition de la latitude et de la longitude

<sup>6</sup> <https://www150.statcan.gc.ca/n1/pub/92-195-x/2011001/other-autre/coord/coord-fra.htm>  
<http://rochetrejoux.pagesperso-orange.fr/EP/Outils/Coordonnees%20GPS.htm>  
<https://blogs.msdn.microsoft.com/ogdifrance/2011/07/13/de-la-go-et-des-maths/>

La latitude est l'angle  $\vartheta$  entre l'axe et le plan équateur. Il s'étend de  $-90^\circ$  au pôle sud à  $90^\circ$  au pôle nord.

La longitude est l'angle  $\phi$  dans le plan équateur. Il a une étendu de  $180^\circ$  ouest à  $180^\circ$  est.

Toutes les distances calculées seront à vol d'oiseau. Pour cela, la fonction suivante est utilisée :

```
float ComputeDist(double latitude1, double longitude1, double latitude2,
                 double longitude2);
```

#### Fonction 10 : ComputeDist

Cette fonction prend en argument la latitude et la longitude des deux positions. Ces données étant en degré décimal, il faut les transformer en radian. Pour cela, il faut les multiplier par  $\pi$  puis diviser par 180. Ensuite voici l'équation utilisée par la Fonction 10 :

$$distance = R_{Terre} * Acos(\cos(latitude1) * \cos(latitude2) * \cos(longitude2 - longitude1) + \sin(latitude1) * \sin(latitude2))$$

#### Équation 3 : Distance entre 2 coordonnées GPS

Le  $R_{Terre}$  est le rayon de la terre et vaut  $6'371'008$  m.

Pour vérifier l'exactitude de cette équation, voici un tableau qui compare pour plusieurs positions leur distance obtenue à l'aide de l'équation avec celle obtenue avec Google Maps :

Latitude A	Longitude A	Latitude B	Longitude B	Calcul [m]	Google Maps [m]	Erreur [%]
46.220173	7.305238	46.219614	7.303627	138.74	137.62	0.81
46.219514	7.303627	46.241002	7.360544	4982.3	4990	0.15
46.220173	7.305238	46.241242	7.360131	4828.8	4830	0.024
46.105942	7.080010	46.227584	7.358535	25358	25470	0.44

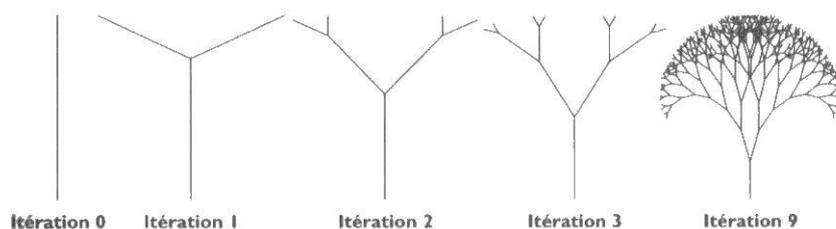
Tableau 19 : Calcul de la distance

Comme le montre le tableau ci-dessus, l'erreur relative entre la valeur de distance donnée par google Maps et celle calculé est moins de 1%. Ce qui est tout à fait raisonnable.

### 3.6.3 Chemin idéal

Le chemin idéal est celui qu'un individu met le moins de temps à parcourir. Parfois il existe des chemins différents qui permettent à un individu de se rendre de son arrêt de départ à son arrêt de destination et ce n'est pas toujours celui qui a la plus courte distance qui est le plus rapide. Pour trouver le parcours le plus rapide, une fois que tous les chemins possibles ont été trouvés, il faut regarder avec la table horaire pour avoir le temps de trajet prévu ainsi que le temps d'attente en cas de correspondance.

Tout d'abord pour déterminer les chemins possibles plusieurs fonctions sont utilisées sur le principe de la récursivité qui est une démarche qui fait référence à l'objet même de la démarche à un moment du processus.

Figure 29 : Exemple de récursivité<sup>7</sup>

En informatique, lorsque deux fonctions ou plus s'appellent entre elles, il s'agit d'une récursivité croisée. Voici les deux fonctions récursives utilisées :

```
int DetermineChemins(int id_arret_depart, int id_arret_destination,
                    struct CHEMINS *chemins)
```

Fonction 11 : DetermineChemins

```
int DeterminePortionChemin(int id_arret_depart,
                          int id_arret_destination, int *nBif, char **portionChemin)
```

Fonction 12 : DeterminePortionChemin

La Fonction 11 construit le chemin en cours depuis l'arrêt de départ donné en argument en additionnant les portions de chemins données par la Fonction 12 jusqu'à l'arrêt de destination. Lorsqu'elle reçoit une portion de chemin, 4 cas sont possibles :

Cas	Description
<b>1 : Destination atteinte</b>	Le chemin est utilisable donc il est ajouté à la liste de chemins possibles puis il passe au chemin suivant s'il y en a.
<b>2 : Bifurcation</b>	Le chemin actuel est copié et vérifié si un des arrêts de la bifurcation est l'arrêt de destination. Si c'est le cas alors il est ajouté à la liste de chemins possibles puis il passe au suivant.
<b>3 : Chemin normal</b>	La portion de chemin est ajoutée au chemin en cours
<b>4 : Chemin terminé</b>	Le chemin est terminé sans que l'arrêt de destination soit atteint. Alors il passe au chemin suivant s'il y en a.

Tableau 20 : Différentes possibilités de chemins

Pour chaque portion de chemins donnés, l'arrêt de destination est vérifié par la fonction suivante :

```
int Presence(char *listCheminActuel, struct LIGNE_ARRET *ligneArret)
```

Fonction 13 : Presence

Si elle ne se trouve pas déjà sur le chemin en cours, elle est ajoutée. Dans le cas contraire, elle ne sera pas comptabilisée et le chemin en cours est terminé.

<sup>7</sup> <http://imagesbiogeolfxm.free.fr/divers/original/ARBRE%20FRACTAL.html>

Voici un exemple d'utilisation des fonctions pour le déplacement de l'individu 0 pour les lignes surlignées en jaune dans les Tableau 14, Tableau 15 et Tableau 16 :

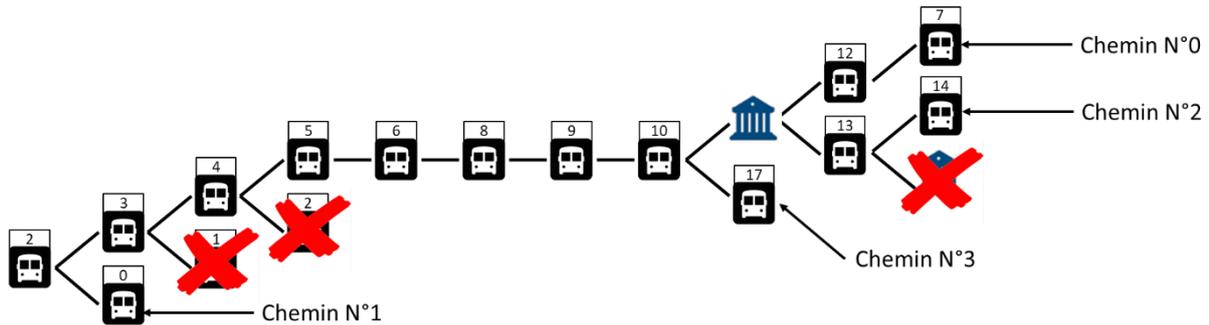


Figure 30 : Exemple de chemins

L'icône bleue représente la gare et les chemins impossibles sont tracés en rouge. Une fois que tous les chemins possibles ont été trouvés, ils sont ajoutés à une liste de chemins possibles. Puis les chemins seront complétés en commençant par le chemin N° 0.

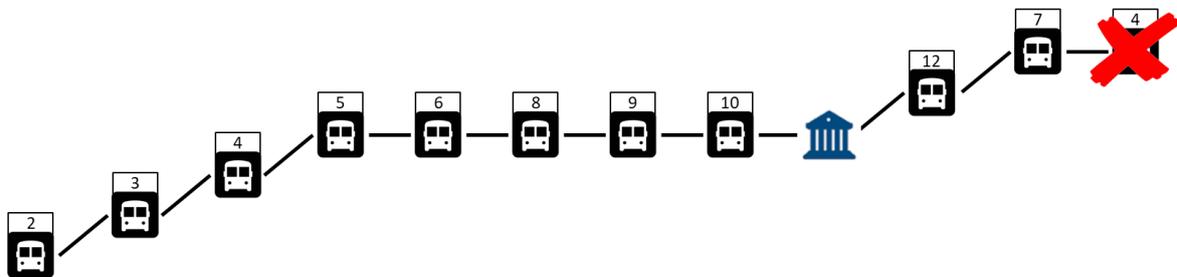


Figure 31 : Chemin N°0

Le chemin N°0 est impossible car avant d'arriver à l'arrêt de destination, l'arrêt numéro 4 est de nouveau présent. (Fonction 13) La fonction va donc passer au chemin suivant qui est le N°1.

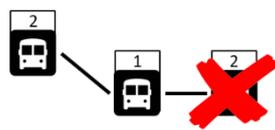


Figure 32 : Chemin N°1

Le chemin N°1 est également impossible alors il passe alors au suivant qui est le N°2.

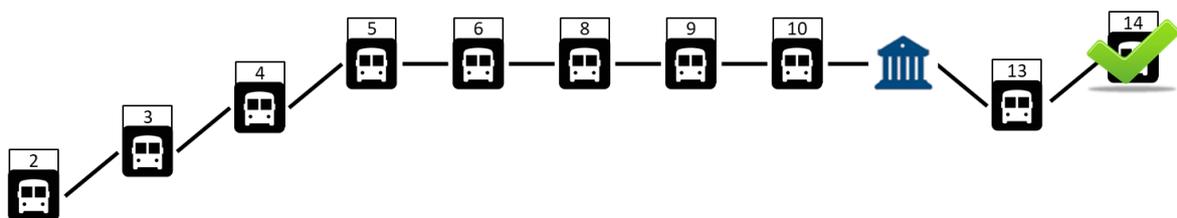


Figure 33 : Chemin N°2

Celui-ci contrairement aux précédents est un candidat pour le parcours de l'individu 0. De ce fait, le chemin N°2 est ajouté à une liste de parcours possibles. S'il y a d'autre chemins possible la fonction passe au suivant qui ici est le N°3.

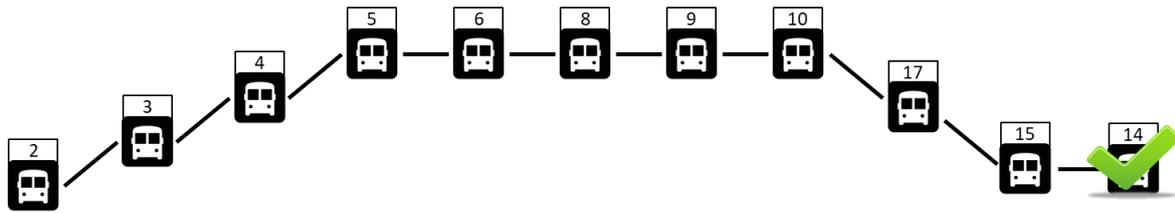


Figure 34 : Chemin N°3

Le chemin N°4 est également un candidat possible pour l’individu. Donc il est également ajouté à la liste de parcours possibles. Comme c’était le dernier chemin possible, la fonction récursive s’arrête là et va retourner une liste avec les deux parcours possibles.

Voici les deux listes de parcours possibles :

id_trajet	id_ligne	id_arret_ depart	id_arret_ arrivee
2	1	2	3
3	1	3	4
4	1	4	5
5	1	5	6
6	1	6	8
7	1	8	9
8	1	9	10
9	1	10	11
16	3	11	13
17	3	13	14

Tableau 21 : Liste de parcours du chemin N°2

id_trajet	id_ligne	id_arret_ depart	id_arret_ arrivee
2	1	2	3
3	1	3	4
4	1	4	5
5	1	5	6
6	1	6	8
7	1	8	9
8	1	9	10
37	8	10	17
38	8	17	15
39	8	15	14

Tableau 22 : Liste de parcours du chemin N° 3

Les deux candidats ont une correspondance. Le chemin N°2 a une correspondance à la gare qui est l’arrêt numéro 11 et le chemin N°3 a une correspondance à l’arrêt numéro 10.

Après avoir listé tous les chemins possibles, il faut en choisir un qui remplit les critères suivants :

- Respecter **heure\_arrivee** du Tableau 8
- Minimiser le temps de trajet

Ces deux conditions dépendent de la table horaire (Tableau 13). Voici la logique programmée pour trouver l’horaire de l’individu :

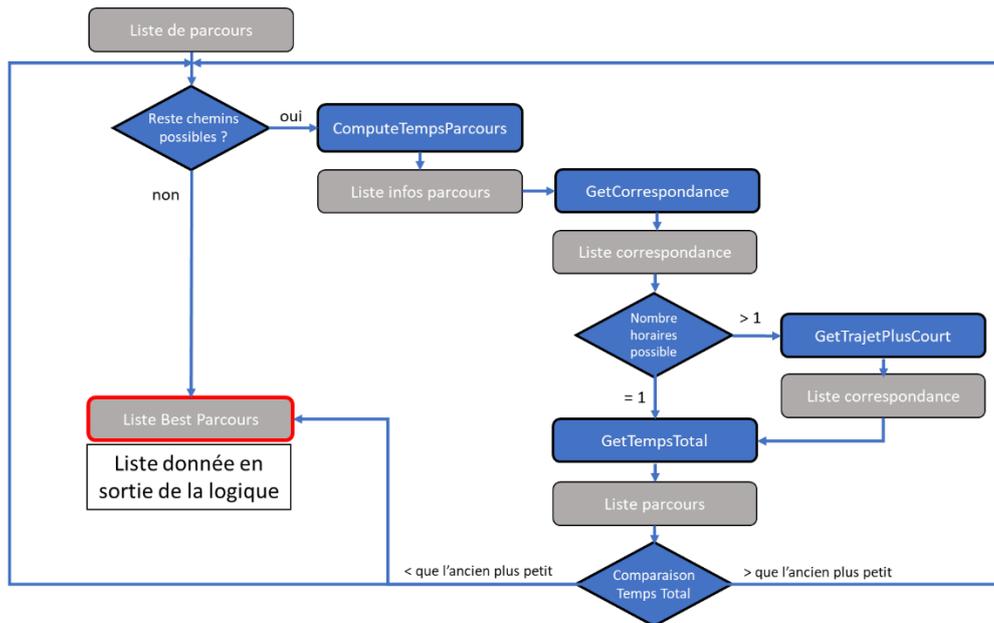


Figure 35 : Logique pour trouver l'horaire idéal

Dans la Figure 35, les cases en bleu sont des fonctions et celles en gris sont des listes retournées par les fonctions. La première liste qui est la liste parcours est celle des Tableau 21 et Tableau 22. La logique va vérifier s’il reste des chemins possibles.

- Si oui : la fonction 15 est appelée
- Si non : la liste Best Parcours est retournée

En reprenant l’exemple précédent pour l’individu 0 voici ce que fait la logique :

```
int Mobility::ComputeTempsParcours(char *chemins, char **listParcours)
```

Fonction 14 : ComputeTempsParcours

Cette fonction va prendre un chemin de la liste de parcours et calculer le temps que passe l’individu dans le bus et il va retourner la liste suivante :

id_ligne	id_arret_	id_arret_d	distance	temps
	depart	estination	Total	Total
1	2	11	5312	960
3	11	14	2200	550

Tableau 23 : Liste infos parcours

Une fois cette liste obtenue, il faut aller chercher dans la table horaire (Tableau 13) tous les horaires correspondant pour l’heure d’arrivée de l’individu. C’est ce que fait la fonction suivante :

```
int Mobility :: GetCorrespondance(char *HeureArrive, int tempsAW_W,
int id_arret_dest, char *listBin, char *listParcours,
char**listCorrespondance)
```

Fonction 15 : GetCorrespondance

Cette fonction retourne une liste avec l'horaire de départ possible. Lorsqu'un changement de bus est nécessaire, l'horaire de départ de la correspondance est également donné. Voici cette liste :

Horaire															
id_horaire	id_engagement	id_bus	id_ligne	id_trajet_ligne	id_arret_depart	id_arret_arrivee	Heure_plantifie_depart	Heure_plantifie_arrivee	Heure_Reel_depart	Heure_Reel_arrivee	Places_Assises	Places_Debouts	Places_Totals	Places_Libres	Statut
61	10	3	3	16	11	13	07:36:00	07:44:00	07:36:00		20	10	30	30	1
65	11	2	3	16	11	13	07:45:00	07:53:00	07:45:00		30	15	45	45	1
35	7	0	1	2	2	3	07:03:00	07:04:30			20	10	30	30	0
44	8	1	1	2	2	3	07:18:00	07:19:30			30	15	45	45	0

Tableau 24 : Liste correspondance

Cette liste peut très bien contenir qu'un seul horaire possible par trajet. Dans ce cas, le programme passe directement à la fonction suivante :

```
int Mobility::GetTempsTotal(char *listCorrespondance, char *listParcours,
                           char *listBin)
```

Fonction 16 : GetTempsTotal

Si la liste parcours contient plus d'un horaire possible, comme c'est le cas ici dans le Tableau 24, il faut effectuer faire une étape supplémentaire qui utilise la fonction suivante :

```
int Mobility :: GetTrajetsPlusCourt(char *listCorrespondance,
                                    char *listParcours, char *listBin)
```

Fonction 17 : GetTrajetsPlusCourt

Cette fonction va choisir l'horaire qui permet à l'individu d'arriver le plus tard possible à destination tout en respectant son **heure\_arrivee** du Tableau 8. Il va faire cela jusqu'à avoir le trajet complet puis retourner la liste suivante :

Horaire															
id_horaire	id_engagement	id_bus	id_ligne	id_trajet_ligne	id_arret_depart	id_arret_arrivee	Heure_plantifie_depart	Heure_plantifie_arrivee	Heure_Reel_depart	Heure_Reel_arrivee	Places_Assises	Places_Debouts	Places_Totals	Places_Libres	Statut
65	11	2	3	16	11	13	07:45:00	07:53:00	07:45:00		30	15	45	45	1
44	8	1	1	2	2	3	07:18:00	07:19:30			30	15	45	45	0

Tableau 25 : Liste correspondance

Cette nouvelle liste est semblable à celle obtenue au Tableau 24 mais elle ne contient qu'un seul horaire par trajet. Pour rappel, la liste du Tableau 24 pouvait elle aussi ne contenir qu'un seul horaire par trajet.

A présent la Fonction 16 pourra être utilisée. Cette fonction va calculer le temps total que passera l'individu dans les transports avec le temps d'attente pour prendre les correspondances. Une fois que ce temps est obtenu, il sera comparé avec celui du chemin précédent.

- S'il est plus petit, il sera gardé en mémoire et servira de référence pour la prochaine comparaison. Le programme va refaire les mêmes étapes pour le chemin suivant de la liste de parcours (*Tableau 22*) à condition qu'il reste encore des chemins possibles dans la liste.
- S'il est plus grand que le précédent, le programme passe directement au chemin suivant qui se trouve dans la liste de parcours (*Tableau 22*) à condition qu'il reste encore des chemins possibles dans la liste.

Lorsque tous les chemins possibles ont été comparés entre eux et en fonction de la table horaire (*Tableau 13*), il n'en restera qu'un. C'est celui qui met le moins de temps pour qu'un individu fasse le trajet depuis son arrêt de départ à son arrêt de destination.

### 3.7 Autres fonctions programmées

Une septantaine de fonctions ont été développées lors de ce projet. Les fonctions les plus importantes ont été expliquées dans le chapitre précédent. Les autres fonctions ne seront pas détaillées dans ce rapport car elles ne sont pas aussi complexes que celle expliquées. Néanmoins, elles sont commentées dans le code avec une explication de ce qu'elles produisent.

### 3.8 Fonctions pour les plots

La librairie<sup>8</sup> utilisée (hbook.cpp et hbook.h) permet de créer des plots.

```
int hbook1_(int id, char *title, int nchx, float xmin, float xmax)
```

Fonction 18 : hbook1\_

Cette fonction permet de créer un histogramme unidimensionnel avec l'argument *nchx* qui définit le nombre de canaux que comportera l'histogramme.

```
int hfill_(int id, float x, float y, float weight)
```

Fonction 19 : hfill\_

Cette fonction permet de remplir un histogramme. La librairie contient d'autres fonctions qui sont également utilisées.

---

<sup>8</sup> Cette librairie a été fournie par mon professeur Gilbert Morand.

## 4. Test et Résultats

### 4.1 Test

La simulation commence avec l'interface utilisateur suivante :

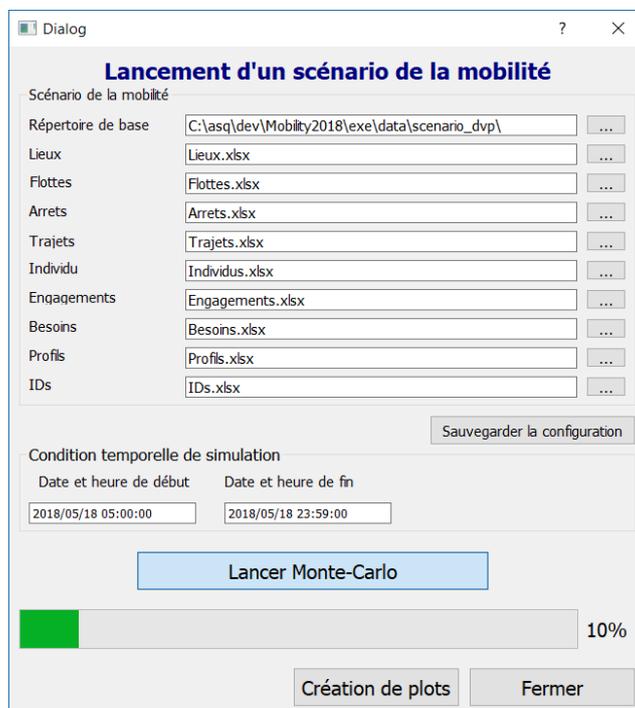


Figure 36 : Interface utilisateur

Cette interface permet de choisir la période temporelle de simulation et le scénario souhaité à l'aide de fichiers Excel, qui contiennent les tables de données. Une fois que tous les paramètres du scénario ont été choisis, en appuyant sur le bouton « Lancer Monte-Carlo » la simulation se lance. Lorsque la barre de progression est complète, en appuyant sur le bouton « Création de plots » les résultats apparaîtront dans le tableau de bord.

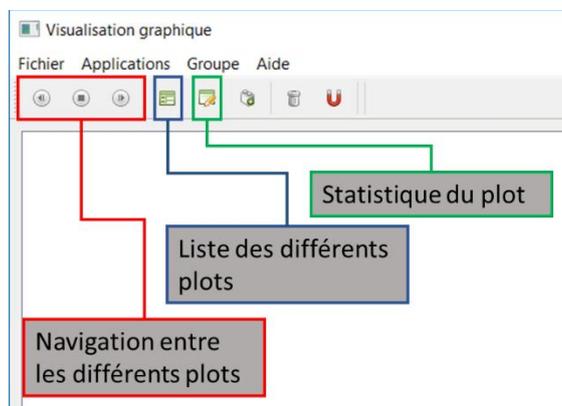


Figure 37 : Tableau de bord

### 4.1.1 Situation et conditions des tests

Les tests effectués se situent dans la région allant de Conthey à Sion et ne représentent pas une vraie demande en mobilité. Cependant, la position des lieux et des arrêts ainsi que les distances entre ces derniers sont des données réelles.

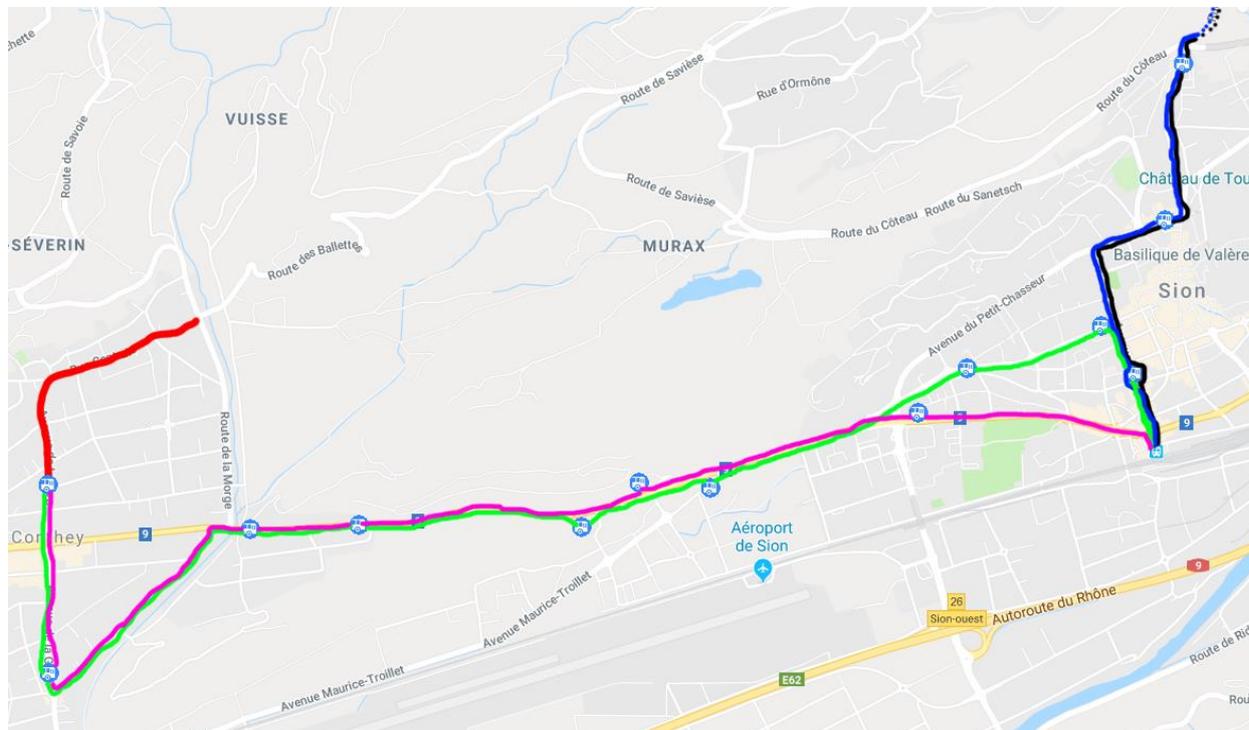


Figure 38 : Carte<sup>9</sup> de la zone de test avec les lignes utilisées

Ces tests ont les conditions suivantes :

- 300 individus habitants à Conthey et travaillants à Sion (Attribution aléatoire)
- Des étudiants qui commencent à 08h00, rentrent à midi et rentrent le soir à 17h00
- Des employés qui commencent à 07h00 et rentrent le soir à 17h00
- Une durée de simulation d'un jour
- Une trentaine d'habitations à Conthey et une dizaine de lieu de travail à Sion
- 15 arrêts de bus dont la Gare
- 4 lignes de bus pour les usagers et une pour le dépôt de bus (en rouge)
- 6 bus qui circulent sur les lignes

## 4.2 Résultats

Certains résultats sont donnés sous forme d'histogramme<sup>10</sup>. En statistiques, un histogramme est un graphique permettant de représenter la répartition d'une variable continue en la représentant avec des colonnes verticales.

<sup>9</sup> Image prise sur google Map

<sup>10</sup> <https://fr.wikipedia.org/wiki/Histogramme>

### 4.2.1 Arrêts

Un des facteurs qui pourrait inciter les personnes à privilégier leur véhicule privé au détriment des transports publics est la distance qui les séparent d'un arrêt de bus. Afin de résoudre ce problème de distance, un placement stratégique des arrêts pourrait encourager les individus à opter pour les transports en commun.

Un des modèles du simulateur qui est le modèle « social » donne l'information de la distance qu'il y a entre les habitations et les arrêts ainsi que la distance entre les lieux de travail et les arrêts bus.

Voici celui qui est obtenue pour les scénarios joué ici :

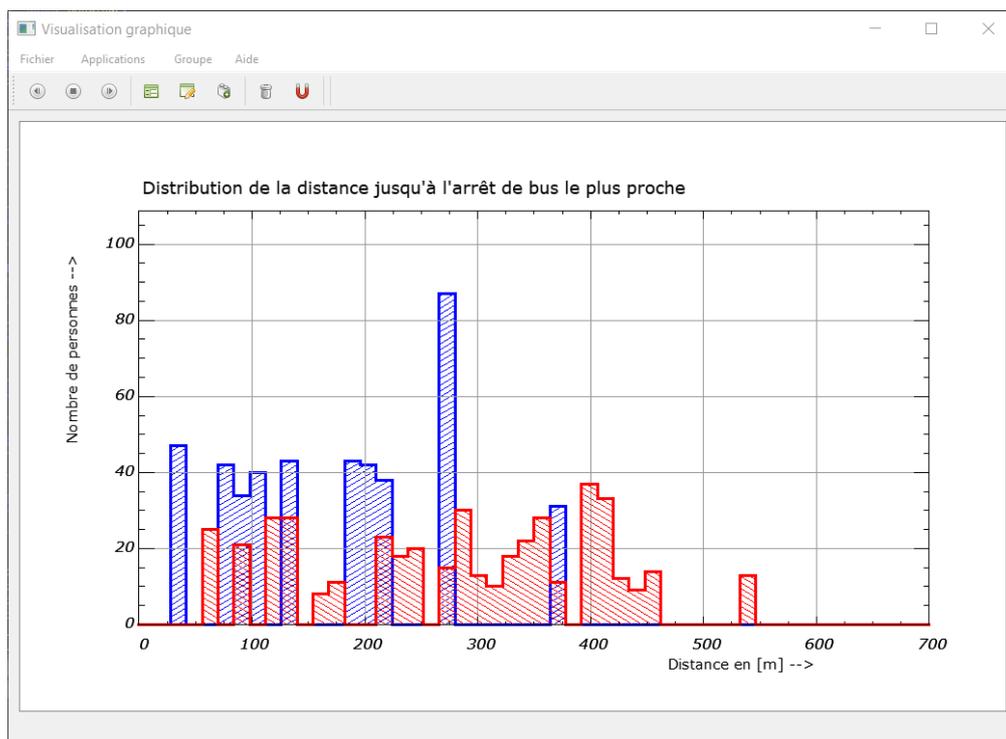


Figure 39 : Histogramme de la distance jusqu'à l'arrêt

- Les colonnes rouges représentent la distance qui sépare les habitations et les arrêts de bus.
- Les colonnes bleues représentent la distance qui sépare les lieux de travail et les arrêts de bus.

A noter que pour les 2 scénarios joués ici les distances jusqu'aux arrêts restent les mêmes car les arrêts n'ont pas été déplacés et les individus habitent et travaillent aux mêmes endroits

### 4.2.2 Parcours

Voici les lignes pour les 2 scénarios pour lesquels les tests ont été effectués :

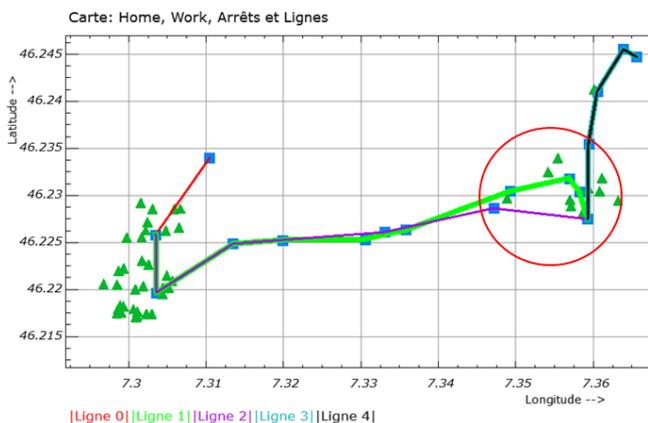


Figure 40 : Parcours du Scénario N°1

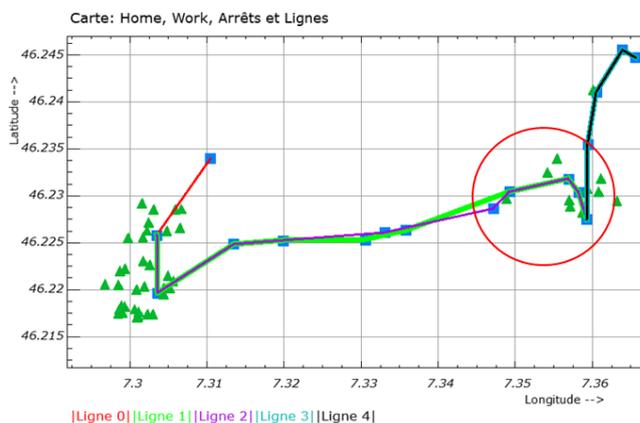


Figure 41 : Parcours du Scénario N°2

- Les triangles verts représentent tous les lieux.
- Les carrés bleus représentent les arrêts de bus

Une modification de la ligne 2 a été réalisée entre les deux scénarios. Cette modification se trouve dans le cercle rouge et il s’agit là de la seule modification effectuée.

A noter que les deux parcours représentés ici sont semblables à la Figure 38.

### 4.2.3 Distance de déplacement

Une autre distance importante à prendre en compte est celle que les individus parcourent en bus, car en l’optimisant, la consommation du bus diminuera.

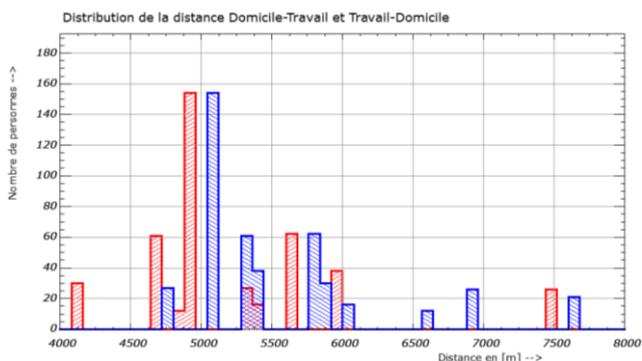


Figure 42 : Distance parcourue par les individus pour le Scénario N°1

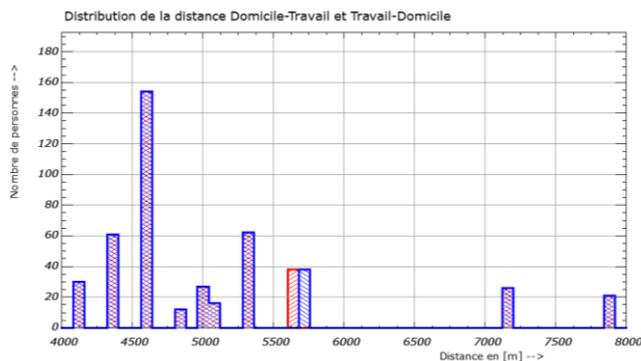


Figure 43 : Distance parcourue par les individus pour le Scénario N°2

- Les colonnes rouges représentent la distance parcourue par les individus pour se rendre à leur lieu de travail
- Les colonnes bleues représentent la distance parcourue par les individus pour rentrer à leur domicile.

Dans le Scénario N°1 quiconque travaillant dans les lieux se trouvant dans le cercle rouge de la Figure 40 doit prendre un bus depuis son arrêt le plus proche jusqu'à la gare puis un deuxième bus jusqu'à son domicile.

Dans le Scénario N°2, la ligne 2 passe cette fois-ci par les arrêts qui se trouvent dans le cercle rouge, de ce fait, les individus qui devaient prendre deux bus pour rentrer n'ont plus que besoin d'un seul.

L'impact de ce changement se voit par la position des colonnes bleues entre les deux graphes.

#### 4.2.4 Temps de déplacement

Un autre facteur important dans le choix du moyen de transport d'un individu est le temps mis pour parvenir à destination. En parvenant à réduire le temps qu'un individu passe dans les transports, y compris le temps d'attente lors des correspondances, la probabilité qu'il opte pour le bus au détriment de son véhicule privé augmentera probablement.

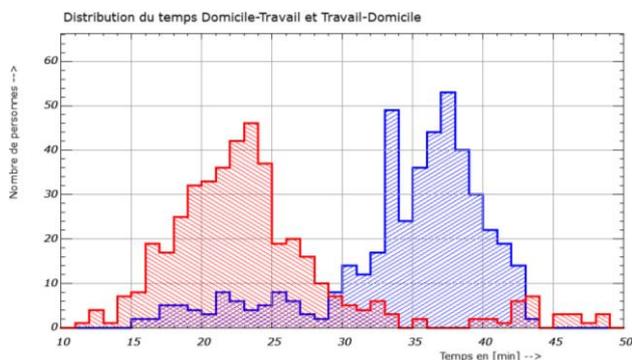


Figure 44 : Temps passés par les individus dans le bus pour le Scénario N°1

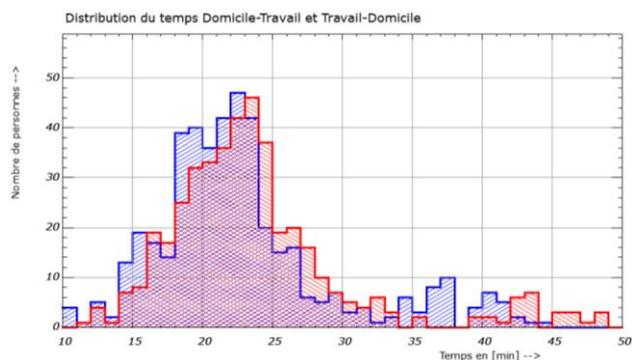


Figure 45 : Temps passés par les individus dans le bus pour le Scénario N°2

- Les colonnes rouges représentent le temps que mettent les individus pour se rendre à leur lieu de travail.
- Les colonnes bleues représentent le temps que mettent les individus pour rentrer à leur domicile.

Le changement de ligne montre sur la Figure 45 une réduction presque d'un facteur 2 du temps passé par les individus pour rentrer chez eux.

Avec ces quelques plots, une amélioration est déjà observable au niveau du temps passé par les usagers dans les transports mais également de la distance parcourue par ces derniers. D'autres plots doivent être réalisés comme par exemple la fréquentation des lignes ou des bus pour avoir une meilleure idée de l'impact du changement de scénario.

## 5. Bus électrique

L'avenir des transports publics comme celui des particuliers passe par l'électricité. Il y a cependant des différences entre les transports publics et les voitures. Premièrement, le fort taux d'usage qui pourra permettre de rentabiliser l'investissement<sup>11</sup> des transports publics mieux que celles des voitures privées grâce à l'écart de prix du diesel et celui de l'électricité. Ensuite, l'expérience acquise avec les systèmes déjà électrifiés comme le métro, le tram et le trolleybus montre l'efficacité et la fiabilité de l'électrification. Finalement, l'autonomie d'une voiture électrique doit être le plus grand possible car le parcours que fait un individu varie tandis que le parcours du bus est connu à l'avance.



Figure 46 : TL Transports publics de la région lausannoise SA<sup>12</sup>

L'objectif est d'avoir des véhicules électriques qui se déplacent sans contact avec une ligne aérienne. Pour cela, il faut stocker quelque part l'énergie dont le bus aura besoin. L'énergie doit être stockée dans des batteries ou des supercondensateurs mais l'objectif est de transporter des passagers et non des tonnes d'accumulateurs. Pour diminuer le nombre d'accumulateurs, la solution est le biberonnage.

Le biberonnage est une succession de recharge partielle d'accumulateurs. Comparant ce système à un marathonien. Ce dernier se ravitaille en chemin pour courir longtemps car le poids est aussi son ennemi. Un autre avantage du biberonnage est le fait que le nombre de batteries peut être adapté à la zone urbaine où le bus circule. En effet, en fonction de la distance qu'il y a entre les arrêts, l'autonomie peut être diminuée car le bus va être rechargé. Toutefois, il faut aussi prendre en compte d'autres éléments du bus qui vont utiliser la batterie comme la climatisation, les écrans d'informations ou encore les prises électriques qui sont à disposition des usagers. En bout de ligne, les bus peuvent être rechargés plus longtemps et complètement.

Un bus 100 % électrique existe déjà en Suisse à Genève où le bus TOSA<sup>13</sup> parcourt la ligne 23 qui fait 12 kilomètres de long (*Annexe 4 et 5*). Pour mettre en place une ligne de bus électrique avec recharge par biberonnage, il faut que l'emplacement des arrêts offre la possibilité de mettre en place l'installation pour la recharge.

<sup>11</sup> [www.04.abb.com](http://www.04.abb.com)

<sup>12</sup> <http://www.tpsep.ch/avril-mai-juin-ii.html>

<sup>13</sup> <https://www.ge.ch/dossier/bus-tosa-innovation-mobilite-au-service-genevois>



Figure 47 : Recharge par biberonnage Bus TOSA<sup>14</sup>

Les navettes autonomes, comme le projet<sup>15</sup> « SmartShuttle » qui est testé actuellement en Ville de Sion, peuvent également être une alternative intéressante dans des zones urbaines. En effet, une combinaison de ces navettes, qui feraient des circuits en ville depuis la gare (Annexe 6) et des bus qui amènerait les individus depuis la périphérie de la ville jusqu'à la gare, est un cas à étudier en ayant une connaissance de la demande des usagers et de la possibilité offerte par la topographie du lieu.



Figure 48 : Navette autonome du projet « SmartShuttle »<sup>16</sup>

<sup>14</sup> <https://www.ge.ch/dossier/bus-tosa-innovation-mobilite-au-service-genevois>

<sup>15</sup> <https://www.postauto.ch/fr/projet-%C2%ABsmartshuttle%C2%BB-0>

<sup>16</sup> <https://www.growingupwithoutborders.com/blog/uncategorized/welcome-to-the-future-smart-shuttle-sion-switzerland/>

## 6. Améliorations futures

Le simulateur doit être perfectionné avant qu'il ne puisse répondre à toutes les questions liées à la mobilité. Voici une liste de quelques améliorations à apporter :

- Ajouter la possibilité de faire prendre le bus à un individu ailleurs qu'à l'arrêt le plus proche de sa position comme à la gare par exemple (Voir Fonction 6 : ChercheArretPlusProche).
- Ajouter une fonction pour gérer le temps d'arrêt d'un bus à un arrêt
- Ajouter une ligne de train avec des individus à prendre et ramener à la gare.
- Faire rentrer les individus chez eux lorsqu'il termine le travail et non comme actuellement à une heure précise.
- Ajouter d'autres modèles pertinents liés à la mobilité.
- Ajouter la possibilité de comparer directement plusieurs scénarios dans le tableau de bord.
- Ajouter une fonction qui donne la position GPS d'un lieu pour une adresse.
- Ajouter une représentation sur une carte en fonction de l'heure (Annexe 8).
- Ajouté une fonction qui crée une table horaire selon le jour de la semaine.

Afin d'électrifier la mobilité, il est important de connaître en détail les trajets qu'empruntent le bus. Actuellement les trajets se font à vol d'oiseau. Pour avoir le détail des trajets, il faudra utiliser une fonction de rappel, connue également sous le nom de callback, qui va chercher les trajets parcourus par les bus et donner la consommation effective.

Ces informations proviendront d'un travail de Bachelor réalisé par Thomas Moser dont le sujet est « **Simulateur de véhicules électriques : caractérisation et spécificités de la mobilité électrique en Suisse.** ».

Il faut également trouver un moyen de connaître les vrais besoins en déplacements des individus pour remplir les tables de données contenant des données personnelles.

## 7. Conclusion

Actuellement la mise en place d'un horaire de bus est réalisée avec une vague connaissance du besoin en mobilité de chaque usager.

Premièrement, dans la plupart des régions en Valais, les bus ont une cadence allant de 10 à 60 minutes. Lors des heures de pointes, cette récurrence peut être justifiée. Néanmoins lors des heures creuses, il se peut que les bus soient quasiment voire même complètement vide.

En utilisant le logiciel développé, une prévision indicative du nombre de personne à déplacer pourra être réalisée. À l'aide de cette information à disposition et en utilisant stratégiquement la flotte de véhicule disponible, une diminution de la pollution ainsi que des économies pourront être accomplies.

Deuxièmement, les bus suivent actuellement un trajet définit par une ligne contenant un certain nombre d'arrêts consécutifs. Comme les chauffeurs du bus n'ont aucun moyen de connaître au départ de la ligne à quels arrêts ils devront s'arrêter pour faire monter ou descendre des individus, ils sont obligés de suivre leurs itinéraires de base, tout en s'arrêtant en cas de besoins. Parfois, ils font inutilement un détour de plusieurs kilomètres car personne n'est monté ou descendu du bus.

En utilisant le logiciel développé, une prévision indicative des arrêts fréquentés peut être réalisée et avec cette information, les trajets des bus peuvent être flexibilisés pour éviter d'emprunter des trajets inutilement.

Dans les deux cas, si les prévisions disent qu'il n'y a personne à déplacer, les bus peuvent être supprimés pour éviter qu'ils ne roulent pour rien. Un système de bus à la demande peut être proposé en complément pour parer à d'éventuels imprévus.

En réduisant la fréquence des bus et en adaptant les véhicules utilisés, il y aura déjà une nette diminution de la pollution. Pour accentuer d'avantage cette diminution de la pollution, il faudrait électrifier les bus. C'est pourquoi une étude plus approfondie, qui pourra être menée à l'aide du logiciel développé, doit être effectué avec les caractéristiques des trajets selon la topographie de la région.

Finalement, si ce projet est poursuivi, le logiciel développé pourrait certainement, avec quelques améliorations, permettre d'optimiser la mobilité tout en répondant concrètement aux interrogations qui y sont liées.

Sion, le 17 août 2018

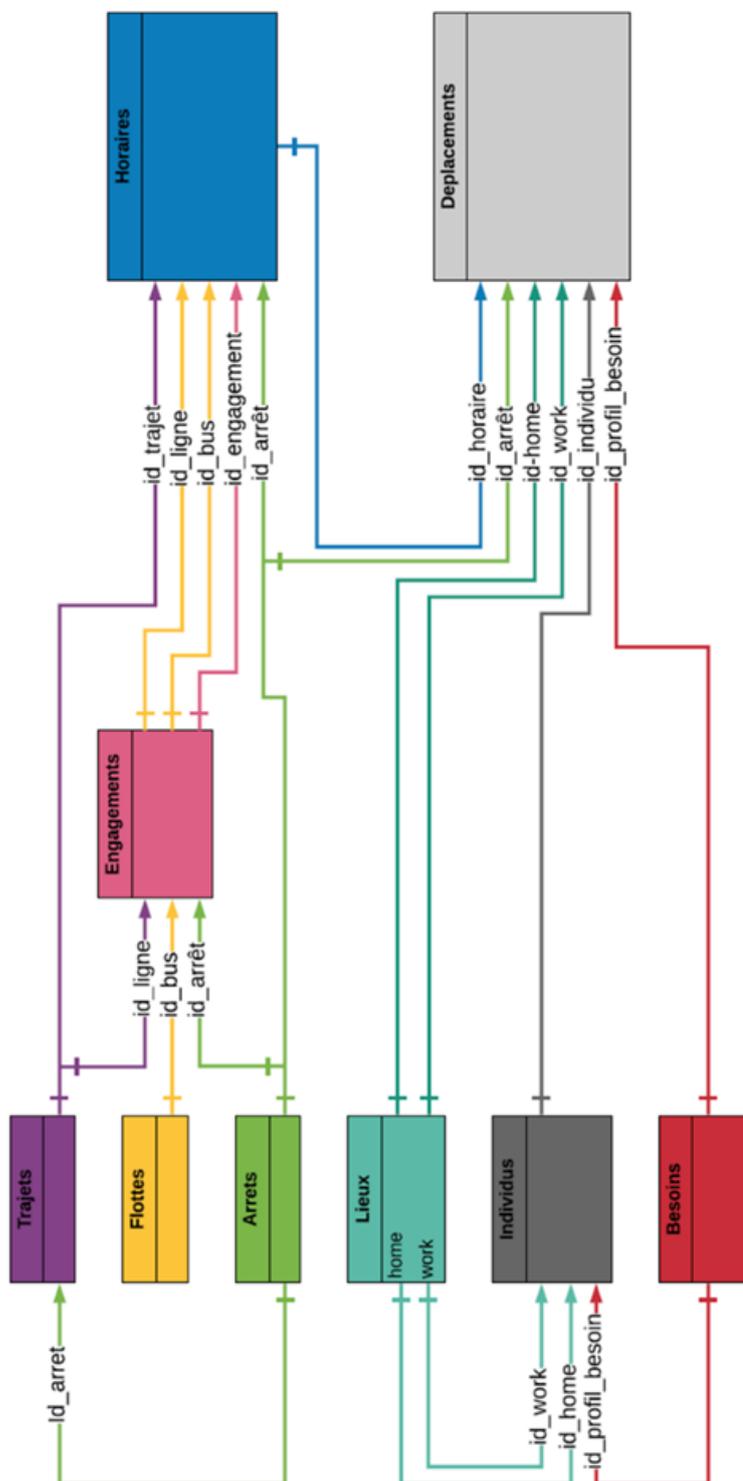
Arumugam Mariatharsan

## 8. Bibliographie

- Sorbonne Université, Master Mathématique et Applications, Spécialité Statistique, Année 2017/2018, Méthode Monte-Carlo par Arnaud Guyader
- HEC Montréal, Haurie, A. & Duguay, C. (1964). Recherche opérationnelle – 5 : Méthode de Monte-Carlo : problèmes d'utilisation. *L'Actualité économique*, 40(3), 577–593. doi:10.7202/1002876ar
- CEA Saclay, Service de Physique de l'Etat Condensé, Année 2008, Méthodes de Monte-Carlo par Michel Roger.
- Cours de théorie de la circulation CIV 6705, La simulation par Luc Pellecuer et Sandrine Poteau
- Nicholas Metropolis, « The Beginning of the Monte Carlo Method », *Los Alamos Science*, n° 15, 1987, p. 125-130
- « RECURSIVITÉ : Définition de RECURSIVITÉ » sur [www.cntrl.fr](http://www.cntrl.fr)

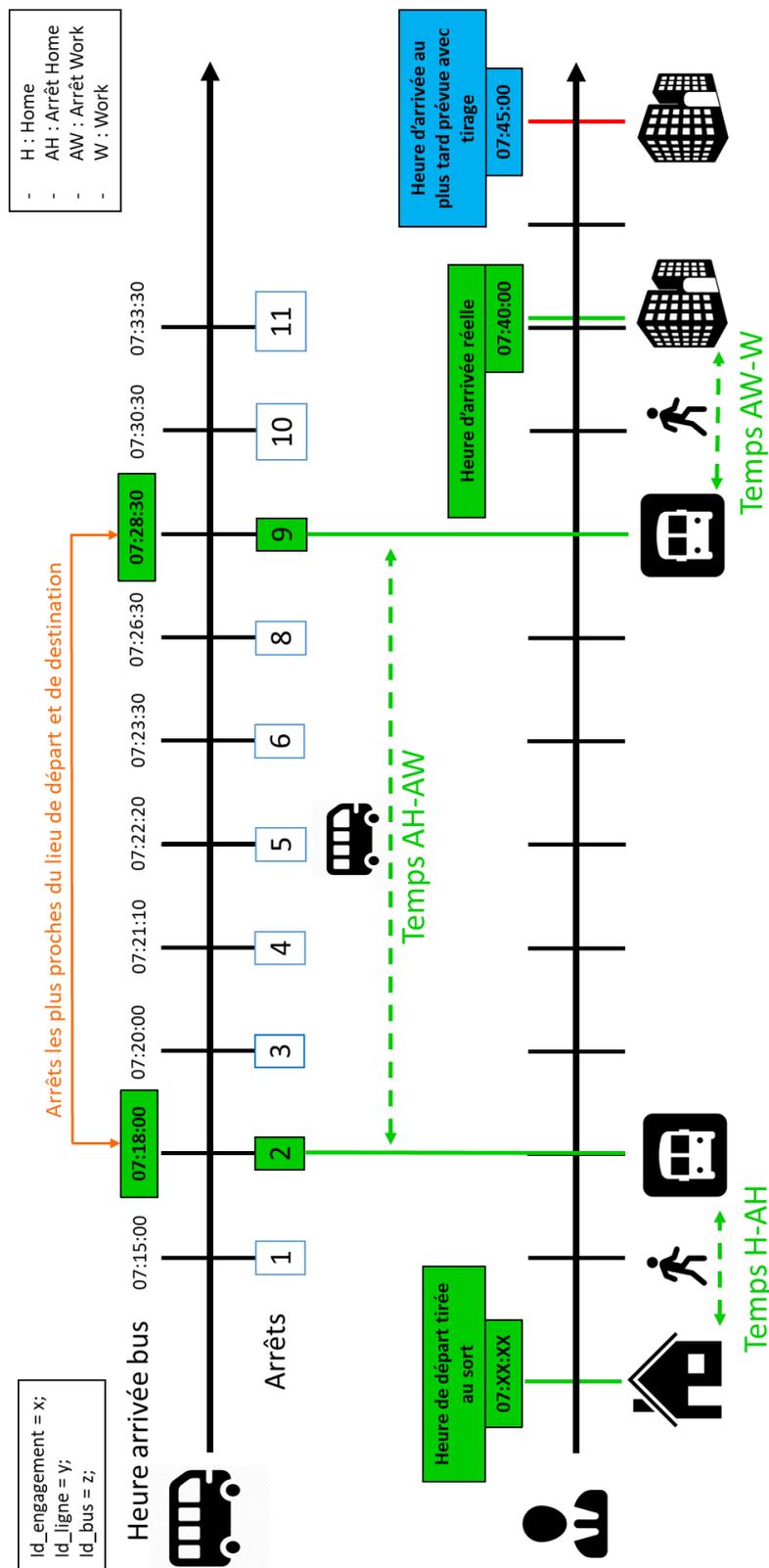
## 9. Annexes

### 9.1 Annexe 1 : Schéma relationnel des tables de données



Annexe 1 : Schéma relationnel des tables de données

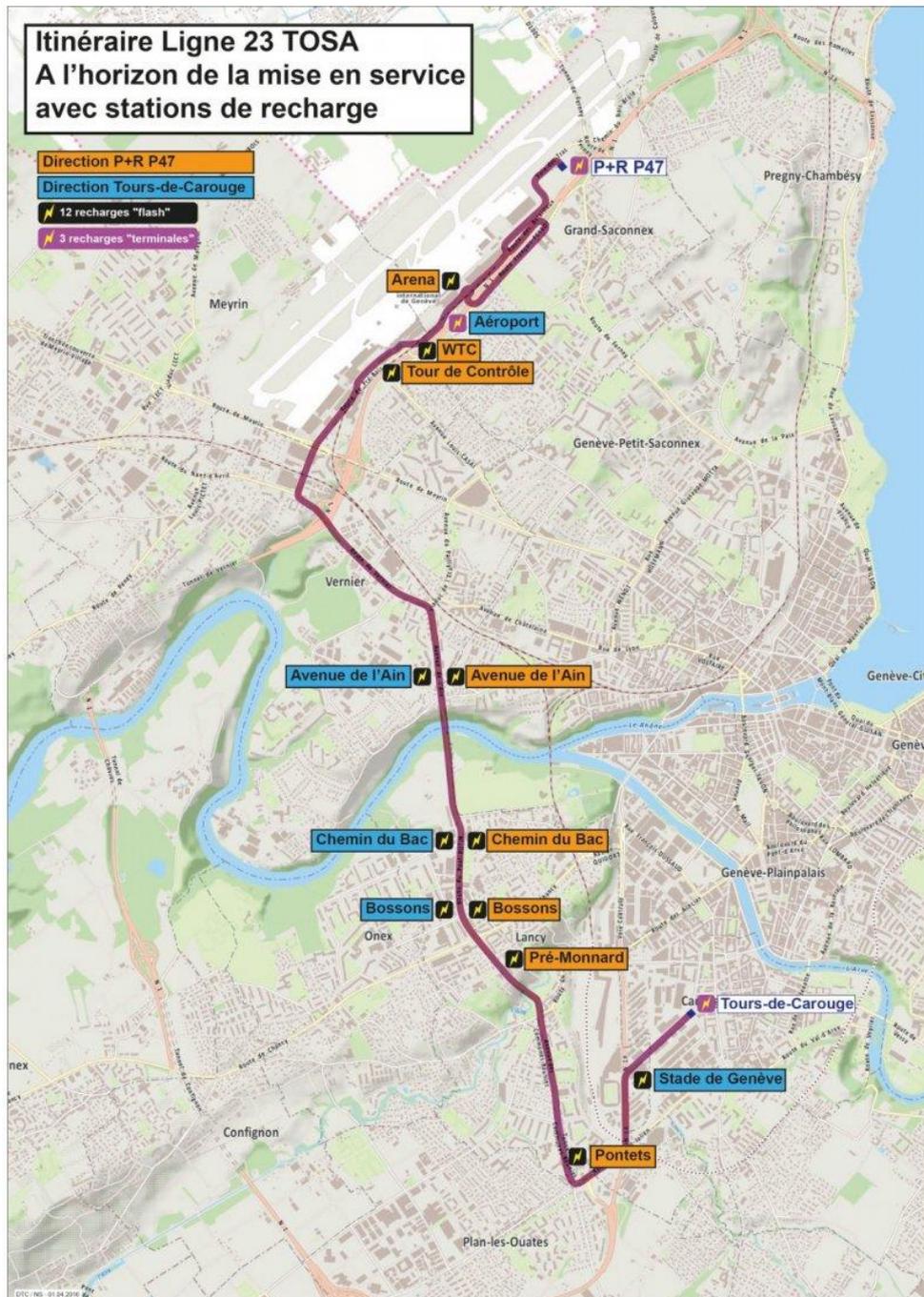
### 9.2 Annexe 2 : Schéma déplacement pour 0 correspondance



Annexe 2 : Schéma déplacement pour 0 correspondance



### 9.4 Annexe 4 : Ligne 23 TOSA



Annexe 4 : Ligne 23 TOSA

## 9.5 Annexe 5 : Informations du Bus TOSA

### Chiffres clés

TOSA – un bus à batteries rechargeables	
15 secondes	technologie de recharge flash aux arrêts
600 KW	puissance de sortie des recharges flash aux arrêts
1 seconde	temps pour se connecter à la station de recharge. Celle-ci est détectée par capteur laser, ce qui rend le processus quasiment instantané.
20 secondes	temps pour se décharger dans la batterie embarquée à la puissance de 400 kW
400 KW	puissance de sortie des recharges aux terminus
4 à 5 minutes	temps de recharge nécessaire au terminal
15% à 30%	de places (voyageurs) en plus
10%	d'économie en consommation énergétique puisque le véhicule est allégé. Le bus transporte des passagers, non des batteries.
10 ans	de durée de vie des batteries
TOSA contribue à améliorer la qualité de l'air	
40%	part des émissions d'oxydes d'azote provenant du trafic routier
33%	part des émissions de particules fines provenant du trafic routier
50%	de la flotte TPG est composée de moteurs diesel, soit plus de 200 véhicules.
0	d'oxyde d'azote produit par un véhicule électrique
0	particule fine liée à la combustion, véhicule
410'000 litres	de diesel économisés sur année par rapport à un bus thermique
1'000 tonnes	de réduction de CO2 par bus par année (comparé au diesel)
TOSA améliore le confort sonore des usagers du bus comme celui des riverains	
60 dbA	au démarrage, ou lors d'une accélération de 20 à 30 km/h, le bruit mesuré est de 60 dbA avec TOSA, soit 10 dbA de moins comparé à un bus articulé diesel
10 dbA	de différence d'émission sonore
50%	de bruit en moins (à l'oreille) lors du démarrage, ou lors d'une accélération
TOSA Ligne 23	
12	véhicules
12	kilomètres de ligne
48	kilomètres de lignes de contact évitées
13	nombre de stations de recharge flash le long du trajet
3	nombre de stations de recharge aux terminus
50	temps de parcours en minutes en heure de pointe du soir
10	un départ toutes les 10 minutes aux heures de pointe à partir de chaque terminus
133	passagers (capacité pour un bus articulé)
10'000	voyageurs par jour

Annexe 5 : Informations du Bus TOSA

### 9.6 Annexe 6 : Parcours des navettes autonomes

# Parcours des navettes autonomes.

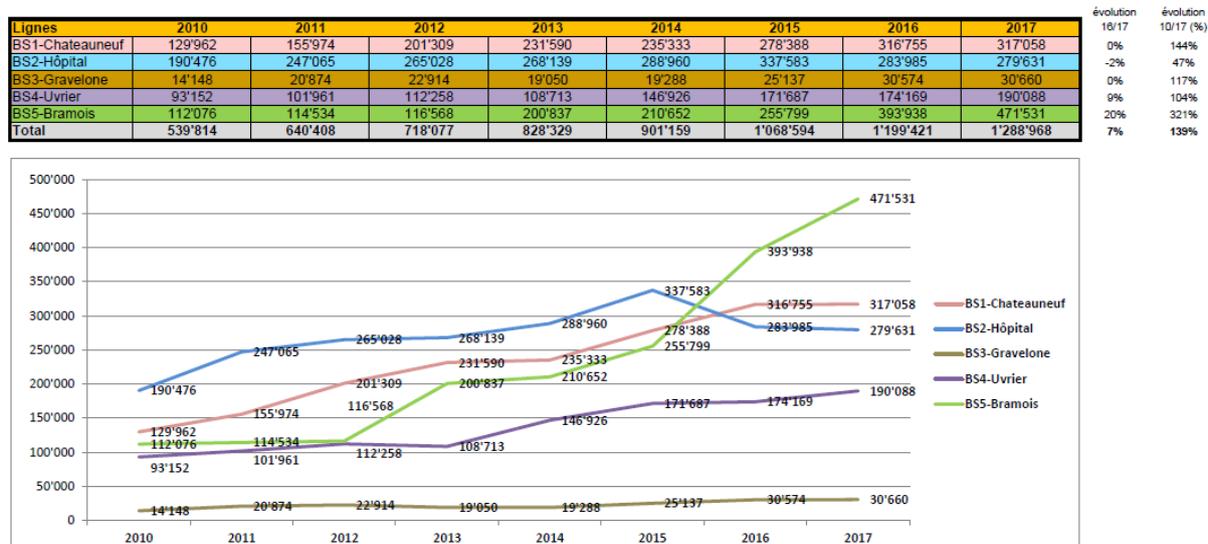


- Trajet actuel
- Extension du trajet
- - - Trajets de déviation



Annexe 6 : Parcours des navettes autonomes

### 9.7 Annexe 7 : Informations du nombre de voyageurs

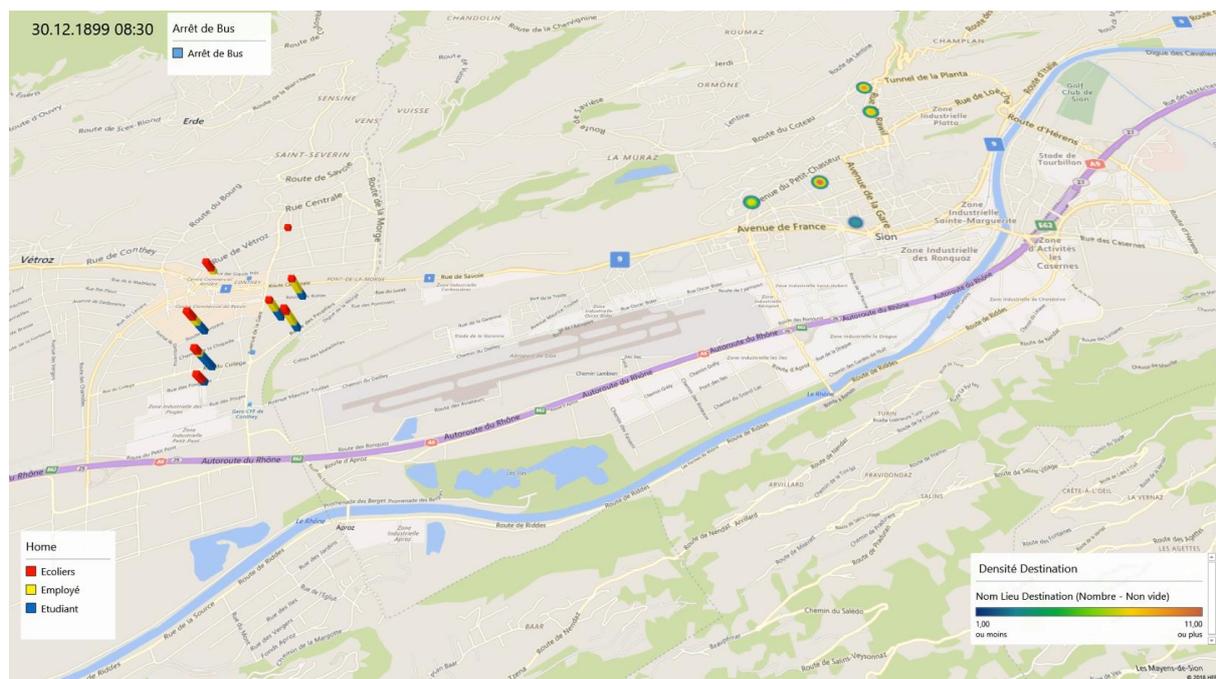


Annexe 7 : Informations du nombre de voyageurs

Ce graphique a été fourni par le Service de l’Urbanisme et de la Mobilité de la Ville de Sion. Il montre uniquement les voyageurs sur une année qui empruntent les BUS SÉDUNOIS.

### 9.8 Annexe 8 : Représentation sur Power Map

Les données fournis par le logiciel développé peuvent être représenté sur Power Map qui est un module de Excel. De cette façon, une vision d’une région permettra de se représenter plus facilement les zones les plus fréquentées.



Annexe 8 : Représentation sur Power Map

## 9.9 Annexe 9 : Installation module xlsx

Un module complémentaire doit être ajouté à Qt-Creator pour qu'il puisse lire les tables de données qui sont fournis en entrée sous le format Excel.

### Installation xlsx dans Qt

#### 1. How to Install

1. Download QtXlsx from VSRonin.
2. Put the source code in any directory you like.
3. Go to top directory of the project in a terminal (cd path/to/your/project).
4. Type:

```
qmake
```

```
mingw32-make
```

```
mingw32-make install
```

#### 2. Conclusion

Now to test if the installation worked, open a project and type this on the .pro:

```
QT += xlsx
```

Now compile it. If the compilation finishes, the module is installed.

Good coding :)

[shareimprove this answer](#)

#### Mise à jour de l'interface

When you create the Makefile by using qmake from the project file, it creates a rule to generate the ui\_mainwindow.h file. However, you can also create it using Qt's uic program from command line:

```
uic mainwindow.ui -o ui_mainwindow.h
```

-or-

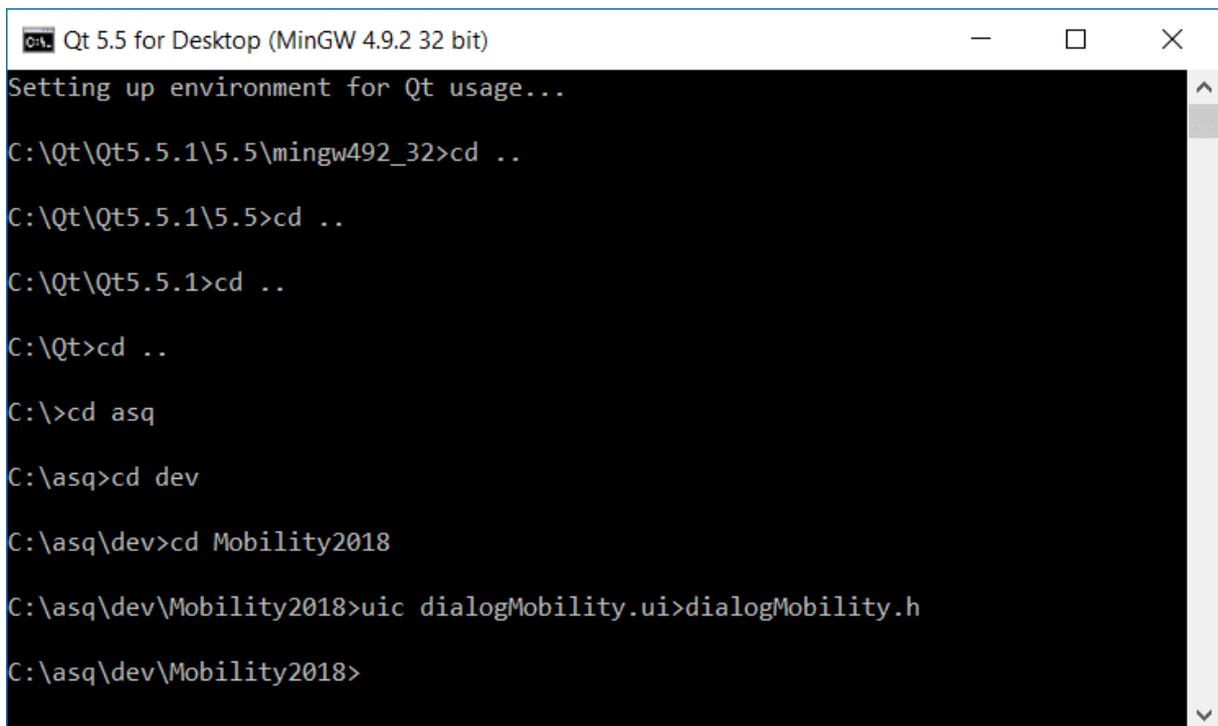
```
uic mainwindow.ui > ui_mainwindow.h
```

Annexe 9 : Installation module xlsx

## 9.10 Annexe 10 : Mise à jour de l'interface utilisateur

Lors d'une modification de l'interface utilisateur, il faut faire la manipulation suivante dans la console de qt :

1. Ouvrir la console version Qt
2. Aller à l'emplacement du fichier à copier
3. Taper la commande **uic** suivi du nom du fichier à copier (exemple : mainwindow.ui > ui\_mainwindow.h)
4. Mettre le fichier créé dans le bon dossier
5. Tout recompiler dans Qt



```
Qt 5.5 for Desktop (MinGW 4.9.2 32 bit)
Setting up environment for Qt usage...
C:\Qt\Qt5.5.1\5.5\mingw492_32>cd ..
C:\Qt\Qt5.5.1\5.5>cd ..
C:\Qt\Qt5.5.1>cd ..
C:\Qt>cd ..
C:\>cd asq
C:\asq>cd dev
C:\asq\dev>cd Mobility2018
C:\asq\dev\Mobility2018>uic dialogMobility.ui>dialogMobility.h
C:\asq\dev\Mobility2018>
```

Annexe 10 : Console de Qt