

**Titre:** Méthode de calibration des logiciels de microsimulation routière à l'aide de l'optimisation sans dérivées  
Title: l'aide de l'optimisation sans dérivées

**Auteur:** Laurent Gauthier  
Author:

**Date:** 2015

**Type:** Mémoire ou thèse / Dissertation or Thesis

**Référence:** Gauthier, L. (2015). Méthode de calibration des logiciels de microsimulation routière à l'aide de l'optimisation sans dérivées [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. <https://publications.polymtl.ca/1984/>  
Citation:

## Document en libre accès dans PolyPublie Open Access document in PolyPublie

**URL de PolyPublie:** <https://publications.polymtl.ca/1984/>  
PolyPublie URL:

**Directeurs de recherche:** Nicolas Saunier  
Advisors:

**Programme:** Génie civil  
Program:

UNIVERSITÉ DE MONTRÉAL

MÉTHODE DE CALIBRATION DES LOGICIELS DE MICROSIMULATION ROUTIÈRE À  
L'AIDE DE L'OPTIMISATION SANS DÉRIVÉES

LAURENT GAUTHIER

DÉPARTEMENT DES GÉNIES CIVIL, GÉOLOGIQUE ET DES MINES  
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION  
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES  
(GÉNIE CIVIL)  
DÉCEMBRE 2015

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

MÉTHODE DE CALIBRATION DES LOGICIELS DE MICROSIMULATION  
ROUTIÈRE À L'AIDE DE L'OPTIMISATION SANS DÉRIVÉES

présenté par : GAUTHIER Laurent

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. LE DIGABEL Sébastien, Ph. D., président

M. SAUNIER Nicolas, Ph. D., membre et directeur de recherche

M. SPURR Timothy, Ph. D., membre

## DÉDICACE

*À mon oncle Hubert, avec qui j'aurais aimé discuter de la vie polytechnicienne*

## REMERCIEMENTS

J'aimerais d'abord remercier mon directeur de recherche, Nicolas Saunier, d'avoir eu la patience et la gentillesse de m'apprendre à programmer et d'avoir été si disponible tout au long de ma maîtrise. J'aurais difficilement pu demander un meilleur superviseur.

J'aimerais également remercier Sébastien Le Digabel, qui a été une ressource inestimable pour la portion optimisation de ma recherche. Sa passion pour son domaine d'étude est contagieuse. Merci de m'avoir enseigné les rudiments d'un domaine qui me saura probablement toujours utile dans la pratique de l'ingénierie.

Merci à mes amis qui ont souffert mes conversations tournant souvent autour de mon sujet de recherche, principalement dans les moments les plus cruciaux du cheminement, comme l'écriture de ce mémoire. Je vous promets que je vais changer de sujet un peu.

Finalement j'aimerais remercier Vincent C. Allaire qui s'est offert afin de me relire, deux fois plutôt qu'une. Tes commentaires ont été très utiles et appréciés.

## RÉSUMÉ

La modélisation de circulation routière est un outil couramment utilisé en ingénierie des transports afin de prévoir les conséquences de modification au réseau routier ou aux infrastructures de contrôle, ou encore pour évaluer l'impact des fermetures nécessaires aux interventions d'entretien du réseau et de suggérer des mesures de mitigations. Une des familles de logiciels utilisées est celle des logiciels de microsimulation, qui calculent la circulation en décomposant les calculs pour chacun des véhicules présents sur le réseau. Ces logiciels sont flexibles, puisqu'ils permettent aux utilisateurs de choisir les valeurs d'une multitude de paramètres, ce qui permet de les calibrer aux différents comportements routiers observés à travers le monde.

Or effectuer cette calibration n'est pas une tâche facile. Plusieurs raisons expliquent cette difficulté, dont entre autres la complexité de l'interaction entre les différents modèles mathématiques décrivant les conditions routières et le peu d'information que les manuels de l'utilisateur fournissent quant aux rôles des différents paramètres. Cela rend difficile la prédiction de l'impact d'une modification de ces paramètres. Un autre facteur limitant est le grand nombre de combinaisons de valeurs que permet l'ensemble de ces paramètres, qui fait en sorte que la calibration manuelle est peu efficace. De plus, obtenir les données nécessaires à la calibration est souvent un enjeu puisqu'il est couteux et difficile d'accéder aux données adéquates pour la calibration des modèles d'interaction entre les véhicules.

Afin de pallier à ces différents problèmes, une méthode de calibration automatique des logiciels de microsimulation est proposée. La méthode repose sur l'utilisation d'un algorithme d'optimisation sans dérivées capable d'explorer l'espace des paramètres afin de trouver l'optimum local d'une fonction objectif, laquelle doit être déterminée selon les besoins de la calibration. Nous présentons l'application de cette méthode à la calibration des paramètres de conduite, soit les modèles expliquant l'interaction des véhicules entre eux à l'intérieur d'une voie (modèles de poursuite) et les modèles expliquant les décisions d'insertion prises par un conducteur souhaitant accéder à la voie adjacente en présence de véhicules dans cette voie (modèles de changement de voie).

La collecte et le traitement des données spécifiques à ce problème sont discutés et analysés, puisque ce type de modèles demande la collecte de données microscopiques comme des données de trajectoire, soit un des types de données les plus difficiles à obtenir puisqu'elle consiste à suivre plusieurs véhicules simultanément sur une certaine distance. Pour ce faire, l'extraction des

trajectoires par l'analyse vidéo est utilisée et les différentes erreurs inhérentes à cette méthode de collecte sont discutées puisqu'elles doivent être détectées et corrigées avant que les données puissent être utilisées dans le processus de calibration. De ces données de trajectoire, ce sont les temps intervéhiculaires (TIV) qui sont dérivés afin que le processus de calibration puisse, à travers la fonction objectif, comparer les distributions de TIV observées sur le terrain et celles générées par le logiciel de microsimulation. Cette comparaison se fait à l'aide du test de Kolmogorov-Smirnov dont la statistique permet de mesurer la distance maximale entre les fonctions de répartition de deux échantillons, soit ici l'échantillon terrain et l'échantillon simulé.

L'implémentation de la méthode proposée, au travers d'un programme Python écrit à cette fin, permet également de faire cette comparaison sur plus d'un jeu de données à la fois. La fonction objectif retourne alors une valeur prenant en compte la distance observée sur chacun des échantillons testés, permettant ainsi d'effectuer une calibration qui prend en compte un ensemble de conditions de circulation différentes et donc d'éviter de surajuster les paramètres pour des conditions particulières. La fonction objectif discutée prend également en compte deux types de TIV, soit les TIV de suivi, associés aux modèles de poursuite, et les TIV acceptés, associés au modèle de changement de voie. Il est également possible d'intégrer des contraintes que l'algorithme d'optimisation devra respecter.

Cette méthode est appliquée spécifiquement aux modèles du logiciel VISSIM 6.0 et en utilisant l'algorithme d'optimisation du Mesh Adaptive Direct Search, implémenté dans le logiciel NOMAD. La calibration est effectuée pour une vingtaine de paramètres à la fois. Ce cas d'étude a nécessité la formulation de quatre contraintes afin de détecter les simulations où des comportements anormaux émergeaient de certains jeux de paramètres, comme des dépassements au travers du véhicule précédent. Chaque évaluation de jeu de paramètres est testée sur vingt répétitions utilisant des nombres d'amorce aléatoire différents et cet ensemble peut être évalué en deux à cinq minutes. Le budget d'évaluations alloué pour la réalisation de la calibration est de 1000, donnant donc un temps de résolution total entre cinq et onze jours. Les résultats montrent que, pour le cas spécifique des modèles de VISSIM, l'inclusion de TIV acceptés gêne la progression de la calibration, puisqu'aucun des jeux de paramètres testés n'a été en mesure de modifier significativement la statistique du test de Kolmogorov-Smirnov sur cet indicateur. Par contre, les résultats montrent que la méthode proposée fonctionne et peut être utilisée pour calibrer des réseaux à simuler.

L'application de la méthode aux modèles de VISSIM a également fait ressortir le besoin d'améliorer la collecte de données, principalement en travaillant sur la calibration du logiciel d'extraction des trajectoires par analyse vidéo utilisé, mais également en portant une attention particulière au positionnement de la caméra durant la collecte. Ces recommandations sont importantes puisque plusieurs données ont dû être écartées de l'application de la méthode à cause d'un taux d'erreur de détection des véhicules trop élevés, ce qui pourrait être réduit en posant les actions nommées ci-dessus.

Finalement, c'est un outil de calibration fonctionnel qui est proposé. Celui-ci permettra aux praticiens d'effectuer des calibrations complexes pour les besoins de leurs études de circulation. Dans son état actuel, l'outil permet la calibration de VISSIM et l'utilisation de NOMAD, mais il sera aisément étendre ses capacités à d'autres logiciels de microsimulation, à d'autres algorithmes d'optimisation, à d'autres types de données que les données vidéo ainsi qu'à des modèles qui n'ont pas été pris en compte dans ce cas d'étude. Cette extension des capacités de l'outil ouvrira la porte à l'étude de plusieurs questions de calibration comme la comparaison de l'efficacité relative des modèles de microsimulation et des modèles de mésosimulation, soit une autre famille de modèle de simulation qui reposent sur un degré moindre de décomposition des calculs et gagnent en popularité grâce à leurs besoins plus faibles en capacité de calcul.

## ABSTRACT

Traffic simulation is now a commonly used tool in transportation engineering to assess the impacts of proposed infrastructure, traffic control modifications, and of necessary lane, or road closures, associated with construction sites. Traffic simulation software are divided between families of implemented models and one of them, the microsimulation family, is popular to assess the local impacts of network modifications. Software from the microsimulation family rely on models that simulate individual vehicles, their interactions with each other, and the geometric design of infrastructures, throughout their trips in the studied network. These software are highly flexible since they give practitioners the possibility to modify a large number of parameters, and to calibrate the software to the different driving behaviors observed around the world.

However, successfully conducting such a calibration is not an easy task. The complex interactions between the different mathematical models describing the traffic conditions and the low level of information provided to users on these models makes the prediction of the impact of a single parameter change almost impossible. The large number of value combinations for the parameters also renders the manual calibration of the models ineffective. Moreover, data acquisition is often a daunting task since it is costly and difficult to collect the necessary data that can help assess how well vehicular interaction models describe reality.

To help circumvent these difficulties, an automatic calibration procedure for microsimulation software is proposed. The procedure relies on the use of a derivative-free optimization algorithm able to explore the search-space formed by the relevant parameters to find the local optimum of an objective function. The function itself must be tailored to the calibration problem's objectives. An application of this method for the calibration of the driving behavior models' parameters is presented. Such models are used to explain the interaction between vehicles inside a traffic lane (car-following model) and to model the merging onto adjacent lanes (lane-changing model).

The data collection and processing techniques specific to this problem are discussed and analysed since these types of microscopic models require a type of traffic data that is one of the hardest to acquire: trajectory data. To acquire such data, trajectory extraction is performed using an automated video tracking algorithm. Typical tracking errors are discussed since they must be detected and corrected for the data to be usable in the calibration process. From these extracted trajectories, time headways are calculated and used to build the objective function, enabling the comparison of

simulated and observed headway distributions. This comparison is done using the statistic of the Kolmogorov-Smirnov test to measure the distance between the cumulative distributions of both samples.

The proposed method's implementation allows to perform this calibration on several data sets at once. This is made possible by a Python program specifically developed for this task. When using this feature, the objective function then returns a value that takes into account the distance for each of the tested data samples. This allows the calibration to take into account diverse traffic conditions and to avoid overfitting the parameters to a single data set. The discussed objective function allows the inclusion of two types of headways, namely the car-following headways associated with the car-following model and accepted headways associated with the lane-changing model. It is also possible to provide optimisation constraints that the algorithm must respect during the calibration process.

A case study is presented which applies the proposed methodology to the specific case of the driving behavior models of VISSIM 6.0 with the Mesh Adaptive Direct Search, implemented in the NOMAD software, as the optimization algorithm. The calibration is performed using a set of twenty parameters. This case study required the formulation of four optimization constraints to detect and discard the parameter sets leading to simulations where a vehicle is allowed by VISSIM to pass through another vehicle in the same lane. Each parameter set evaluation is tested on twenty repetitions using different random seed numbers and can be evaluated in two to five minutes. The evaluation budget allocated for calibration problem is of a 1000, resulting in a total resolution time between five and eleven days. The results show that, for the specific case of the VISSIM driving behavior models, accepted headways impede the calibration process since no tested parameter sets were able to induce any significant change in the statistic of the Kolmogorov-Smirnov test for this indicator. However, the results show that the proposed procedure is functional and may be used to calibrate microsimulation networks.

The case study also highlighted the need to improve the data collection mainly by working on the calibration of the video tracking algorithm but also by paying particular attention to the positioning of the camera during the data collection. These recommendations are important since many videos had to be discarded from the calibration due to a high rate of errors in the detections of the vehicular trajectories. This rate could be lowered by taking the aforementioned actions.

Finally, it's a fully functional automatic calibration tools that is proposed. This tool will provide practitioners the means to perform the complex calibrations and validations needed for their traffic analysis. In its current state, the tool enables the calibration of VISSIM using NOMAD, but can easily be expanded to work with other microsimulation software, other optimization algorithms, various types of input data, and to work on models not covered in this case study. This extension will in turn open the doors to the study of other important calibration questions such as the relative effectiveness of micro and mesosimulations, another family of traffic simulation software that makes use of a lesser degree of disaggregation of the calculations and is gaining popularity because of smaller needs in computing power.

## TABLE DES MATIÈRES

DÉDICACE.....	III
REMERCIEMENTS .....	IV
RÉSUMÉ.....	V
ABSTRACT .....	VIII
TABLE DES MATIÈRES .....	XI
LISTE DES TABLEAUX.....	XIV
LISTE DES FIGURES .....	XV
LISTE DES SIGLES ET ABRÉVIATIONS .....	XVII
LISTE DES ANNEXES .....	XVIII
GLOSSAIRE.....	XIX
CHAPITRE 1 INTRODUCTION.....	1
1.1 Problématique.....	1
1.2 Objectifs .....	3
1.3 Approche méthodologique .....	4
1.4 Structure du document .....	5
CHAPITRE 2 REVUE DE LA LITTÉRATURE.....	7
2.1 Principes de calibration des logiciels de microsimulation .....	8
2.1.1 Phase de préparation.....	8
2.1.2 Phase de calibration (optimisation).....	13
2.1.3 Phase de validation.....	16
2.1.4 Conclusion.....	17
2.2 Analyse de sensibilité.....	18
2.3 Les algorithmes d'optimisation .....	20

CHAPITRE 3 MÉTHODOLOGIE GÉNÉRALE.....	24
3.1    Procédure globale de calibration .....	24
3.1.1    Schéma .....	25
3.1.2    Un logiciel permettant l'automatisation du processus : CATS .....	27
3.2    Phase de préparation.....	27
3.2.1    Formulation du problème .....	27
3.2.2    Choix de l'algorithme d'optimisation .....	34
3.2.3    Méthode de collecte des données .....	35
3.2.4    Traitement des données vidéos .....	37
3.2.5    Identification des paramètres et analyse de sensibilité.....	44
3.3    Phase de calibration.....	46
3.3.1    Choix de la fonction d'ajustement .....	46
3.3.2    La calibration sur plusieurs sites et vidéos.....	47
3.3.3    La fonction objectif .....	48
3.4    Phase de validation.....	48
CHAPITRE 4 CAS D'ÉTUDE.....	50
4.1    Phase de préparation.....	50
4.1.1    Collecte et traitement des données vidéos.....	50
4.1.2    Construction du modèle VISSIM .....	58
4.2    Phase de calibration.....	60
4.2.1    Formulation des contraintes .....	60
4.2.2    Nombre de répétitions .....	65
4.2.3    Point de départ.....	66
4.2.4    Budget d'évaluation .....	67

CHAPITRE 5 RÉSULTATS .....	68
5.1 Calibration des paramètres .....	68
5.2 Paramètres calibrés à utiliser.....	77
CHAPITRE 6 CONCLUSION ET RECOMMANDATIONS .....	80
6.1 Conclusion générale .....	80
6.2 Limitations de la méthode et pistes de solutions.....	81
6.2.1 Limitations techniques .....	81
6.2.2 Sources d'erreurs sur le calcul des paramètres.....	82
6.3 Perspectives de recherche.....	86
BIBLIOGRAPHIE .....	88
ANNEXES .....	94

## LISTE DES TABLEAUX

Tableau 2-1 : Résumé du processus de calibration .....	18
Tableau 4-1: Informations relatives à la collecte des données .....	53
Tableau 4-2: Importance des véhicules dont le suivi est problématique sur l'ensemble des véhicules détectés dans les vidéos de l'A13.....	57
Tableau 4-3: Seuils associés aux contraintes utilisant le fichier d'erreur de VISSIM.....	64
Tableau 5-1 : Paramètres inclus dans l'étude : valeurs par défaut (point de départ), bornes supérieures et inférieures et valeur calibrée .....	68
Tableau 5-2 : Débit de circulation et résultats de l'évaluation de la fonction objectif pour les paramètres par défauts et les paramètres calibrés pour les vidéos utilisées lors de la calibration et de la validation. .....	70
Tableau 5-3 : Paramètres des modèles Wiedemann 99 et de changement de voie proposés pour les segments autoroutiers en section droite de la région montréalaise. ....	77
Tableau 5-4 : Résultat de la fonction objectif pour les paramètres proposés.....	78

## LISTE DES FIGURES

Figure 3-1 : Schéma de la méthodologie générale .....	25
Figure 3-2 : Illustration des différents types de temps intervéhiculaires .....	29
Figure 3-3 : Illustration du point de calcul des TIV de suivi .....	29
Figure 3-4 : Sortie d'autoroute incluant une voie de décélération dans VISSIM .....	31
Figure 3-5: Arbre de classification des changements de voies, utilisé pour classifier chacun des changements de voie d'une trajectoire dans l'ordre chronologique inverse en commençant par le dernier.....	32
Figure 3-6 : Exemple de classification des changements de voies .....	33
Figure 3-7 : Installation du système de collecte vidéo sur un fût d'éclairage du Ministère des Transports du Québec aux abords de l'Autoroute A20 dans l'Ouest de l'île de Montréal ....	36
Figure 3-8 : Illustration du bruit enregistré sur la position du centre de masse d'un objet engendré par la perte et le gain de pixels d'une image à l'autre .....	39
Figure 3-9 : Illustration de l'amplification de l'erreur de parallaxe engendrée par un centre de masse détecté sur le haut d'un camion.....	40
Figure 3-10 : Exemple d'un camion détecté très haut sur la boîte de sa remorque, engendrant une augmentation de l'erreur de parallaxe .....	40
Figure 3-11 : Illustration de la cause des vitesses trop élevées calculées pour les objets ayant un grand effet de parallaxe dû à un suivi des points éloignés du sol .....	42
Figure 3-12 : Assignation aux voies des véhicules détectés durant les 5 premières minutes de la vidéo A13-10.....	44
Figure 3-13 : Illustration de la statistique D du test de Komolgorov-Smirnov.....	46
Figure 4-1 : Les différents lieux de collectes vues selon la caméra déployée .....	51
Figure 4-2 : Vue aérienne des différents sites de collecte, emplacement et champs de vision approximatif des caméras.....	52

Figure 4-3 : Débit observé (et écart relatif) dans les cinq premières minutes des vidéos de l'A13 selon un comptage manuel et le comptage automatique de Traffic Intelligence .....	54
Figure 4-4 : Débit observé (et écart relatif) dans les dix premières minutes des vidéos de l'A13 selon un comptage manuel et le comptage automatique de Traffic Intelligence .....	55
Figure 4-5 : Vitesses moyennes et médianes observées durant les cinq premières minutes des vidéos de l'A13, après réduction de 14 % .....	55
Figure 4-6 : Diagramme fondamental tracé pour le débit et la vitesse moyenne réduite de 14% des 5 premières minutes des vidéos, par intervalle de 20 minutes, pour la section étudiée de l'A13 selon les données expérimentales.....	57
Figure 4-7 : Modèle VISSIM de l'Autoroute 13 .....	59
Figure 4-8 : Distribution cumulée des vitesses désirées telle qu'entrée dans le modèle VISSIM de la A13 .....	60
Figure 4-9 : Exemple d'un carambolage causé par un mauvais jeu de valeurs pour les paramètres de modèle de poursuite Wiedemann 99 .....	62
Figure 4-10 : Diagramme espace-temps de la première voie de circulation du réseau de l'A13 montrant des véhicules passant au travers du véhicule les précédant .....	63
Figure 4-11 : Détermination du nombre de répétitions à effectuer selon une méthode de stabilisation de la valeur de la fonction objectif .....	66
Figure 5-1 : Visualisation de l'espace des paramètres exploré par NOMAD .....	69
Figure 5-2 : Distributions cumulées des TIV de suivi pour les vidéos utilisées lors des étapes de calibration (A13-06, A13-11, A13-12 et A13-13) et de validation (A13-10 et A13-25).....	74
Figure 5-3 : Distributions cumulées des TIV acceptés pour les vidéos utilisées lors des étapes de calibration (A13-06, A13-11, A13-12 et A13-13) et de validation (A13-10 et A13-25).....	75
Figure 5-4 : Visualisation de la progression des différentes composantes de la fonction objectif en fonction du nombre de jeux de paramètres évalués par NOMAD .....	76
Figure 6-1 : Exemple de camions dans la voie du centre de l'A15 masquant la vision d'une autre voie de circulation. ....	84

## LISTE DES SIGLES ET ABRÉVIATIONS

FHWA	Federal Highway Administration
MADS	Mesh Adaptive Direct Search
TIV	Temps intervéhiculaire
ANOVA	Analyse de variance ( <i>ANalysis Of Variance</i> )
CATS	CAlibration of Traffic Simulations

## **LISTE DES ANNEXES**

Annexe A – Fonctionnement de CATS.....	94
Annexe B – Description des paramètres de vissim .....	97

## GLOSSAIRE

Calibration	Processus visant à trouver les valeurs d'un ensemble de paramètres d'un modèle qui optimisent la fonction objectif pour un ensemble d'indicateurs et d'observations.
Contrainte	Relation ou condition devant être respectée par le processus de calibration pour un ensemble d'indicateurs, de paramètres et de fonctions d'ajustement.
Évaluation	Pour un jeu de paramètre donné, résultat de la fonction objectif et du respect des contraintes lui étant associé et permettant à l'algorithme d'optimisation de quantifier l'intérêt du jeu de paramètres courant par rapport aux autres jeux de paramètres testés lors du processus de calibration.
Fonction d'ajustement	Fonction comparant les indicateurs mesurés dans la réalité et générés par un ensemble de simulations (mesure l'ajustement ou l'adéquation du modèle à la réalité par l'intermédiaire des indicateurs).
Fonction objectif	Fonction décrivant la capacité du modèle à reproduire le phénomène modélisé (donne une valeur unique de l'adéquation du modèle mesurée à l'aide d'un ensemble de fonctions d'ajustements).
Indicateurs	Ensemble de variables caractérisant le phénomène (dans notre cas, la circulation) pouvant être observées dans la réalité et générées par la simulation du modèle pour comparaison par la fonction d'ajustement.
Modèle	Représentation simplifiée de la réalité, exprimée dans notre cas sous forme mathématique ou informatique déterminée par un

ensemble de paramètres, permettant de décrire et de reproduire un phénomène physique déterminé.

Observations	Ensemble des données récoltées sur le terrain et servant de comparatif aux résultats simulés dans les processus de calibration et de validation. Elles doivent être caractérisées par les mêmes indicateurs que les résultats simulés.
Paramètres <i>Aussi :</i> <i>variables d'optimisation</i>	Ensemble des variables d'un modèle dont la valeur peut être ajustée. Selon les cas, certains paramètres peuvent être directement observés et fixés (dans notre cas de la simulation de la circulation, des paramètres géométriques du réseau). Les autres ont un effet imprévisible sur les résultats du modèle qui ne permet pas de les déterminer directement.  Dans la littérature mathématique, on nomme <i>variables d'optimisation</i> les variables pouvant être ajustées.
Répétitions	Dans le cas d'un modèle utilisant des processus stochastiques, ensemble de simulations du modèle obtenues en utilisant des valeurs d'amorce aléatoire distinctes.
Simulation	Pour un ensemble de valeurs des paramètres, résultats de prédiction d'un phénomène par le modèle. Dans le cas de la calibration d'un logiciel de microsimulation routière, la simulation est une composante de l'évaluation d'un ensemble de valeurs des paramètres.
Validation	Mesure de l'adéquation du modèle calibré (c'est-à-dire dont les paramètres ont les valeurs identifiées lors de sa calibration) à la réalité avec un ensemble d'observations différent de celui utilisé en calibration (les indicateurs utilisés en calibration et validation pourraient aussi changer).

## CHAPITRE 1 INTRODUCTION

Ce projet de recherche vise la calibration et validation des logiciels de simulation de la circulation routière. L'objectif principal est de développer une méthodologie permettant d'effectuer des calibrations et d'ensuite utiliser cette méthodologie afin de calibrer les paramètres de comportement routier afin de reproduire la circulation autoroutière de la région métropolitaine de Montréal à l'aide du logiciel VISSIM 6.0.

### 1.1 Problématique

L'utilisation des logiciels de simulation routière est maintenant très répandue pour l'étude des projets ayant un impact sur la circulation routière. Ces logiciels permettent de reproduire des comportements des usagers de la route et phénomènes observés sur le réseau routier. Parmi les logiciels disponibles, ceux dit de microsimulation visent à reproduire les comportements individuels des différents véhicules sur un réseau en prenant en compte leur interaction entre eux et avec le réseau routier. Cette famille de logiciel est celle qui nécessite le plus de ressources de calcul dû au grand nombre de calculs nécessaire pour la prise en compte de chacun des véhicules et des interactions qu'ils ont avec leurs voisins et les éléments du réseau.

Une multitude de logiciel de microsimulation sont ainsi disponibles afin de modéliser des problèmes de circulation, dont entre autres AIMSUM, MITSIM, PARAMICS, SUMO et VISSIM. À la Direction des transports de la Ville de Montréal, le logiciel VISSIM est utilisé afin de répliquer et de simuler les conditions de circulations complexes entourant l'implantation d'une nouvelle infrastructure ou les conditions de circulations engendrées par des entraves nécessaires à la réalisation d'un projet (FHWA, 2004a; PTV, 2013).

Le logiciel VISSIM, tout comme les autres logiciels de microsimulation, reproduit les différents aspects de la circulation en modélisation les interactions individuelles des véhicules et en se basant sur une affectation individuelle des véhicules parcourant la modélisation. Une multitude d'aspects sont pris en compte tel que le réseau routier, les systèmes de contrôle de la circulation, les modèles comportementaux de conduite et les choix de destination des véhicules (FHWA, 2004b), qui sont décrits par plusieurs modèles mathématiques fonctionnant en même temps en s'influençant les uns les autres. Les différents aspects sont contrôlés par une multitude de paramètres pouvant être ajustés par l'usager. Cela donne une grande flexibilité au logiciel puisque l'utilisateur peut ainsi

adapter les modèles à des situations différentes. Par exemple, nous pouvons penser que les comportements de conduite pourraient différer entre des régions différentes, ou encore que l'interaction entre deux véhicules se ferait différemment dépendamment du type de route. Dans VISSIM, il suffit alors de créer des classes de routes ou de véhicules différents et d'adapter les comportements sur ces tronçons pour qu'ils reflètent la réalité locale.

Toutefois, cette flexibilité apporte une grande complexité puisque les valeurs des différents paramètres doivent alors être connus pour la zone à l'étude afin d'assurer une validité des solutions étudiées. Une calibration, consistant à rapprocher les résultats simulés des données terrain, est alors nécessaire. La calibration permet ainsi aux modèles de s'adapter à des conditions différentes de circulation, causées soit par des variations dans les aménagements ou par des comportements de conduite différents. De plus, les logiciels de microsimulation ne peuvent prendre en compte l'ensemble des facteurs pouvant affecter la circulation : chacun de leurs modèles étant une simplification d'un phénomène complexe. La calibration permet de minimiser une portion de l'erreur causée par la simplification en intégrants dans les paramètres disponibles du modèle l'effet des facteurs non inclus dans celui-ci.

La difficulté de la calibration repose sur le fait que les paramètres interagissent entre eux à l'intérieur de leurs modèles respectifs, et que les modèles interagissent également l'un avec l'autre. L'espace formé par ces paramètres est très grand et le nombre de combinaisons possibles est alors immense. De plus, les modèles sont souvent peu ou pas expliqués par le développeur du logiciel, limitant la capacité d'analyser quels paramètres sont importants pour simuler une situation particulière.

La disponibilité des données nécessaires à la calibration peut également être un obstacle à la réalisation de l'entreprise soit par le coût ou la difficulté de leur acquisition (Hellinga, 1998). De plus, leur qualité peut affecter les résultats obtenus au terme du processus de calibration et remettre en cause leur potentielle utilisation dans des modélisations autres que celle reproduisant les données.

Heureusement, l'avancée dans les algorithmes d'extraction des trajectoires véhiculaires à partir de données vidéos, comme par exemple Traffic Intelligence (Saunier & Sayed, 2006) ou Urban Tracker (Jodoin, Bilodeau, & Saunier, 2014), rendent cette collecte plus aisée. Ce type de données est très riche pour calibrer nombre des modèles des logiciels de microsimulation puisqu'elles

contiennent l'information de la position en fonction du temps de tous les véhicules passant dans le champ de vision de la caméra. A la différence des données macroscopiques comme les comptages et les temps de parcours utilisées typiquement pour la calibration, il devient alors possible d'étudier les interactions dynamiques entre les véhicules que devraient reproduire les modèles de simulation.

La calibration peut aussi être un processus très long puisque, bien que simple en essence, l'application des techniques d'optimisation peut rapidement devenir complexe (Hellinga, 1998). Le processus de calibration doit donc être bien documenté afin que les prochains utilisateurs puissent réutiliser les valeurs avec confiance ou choisir en toute connaissance de cause de les modifier pour une situation donnée (PTV, 2013).

## 1.2 Objectifs

Pour la région de Montréal, la connaissance des valeurs des paramètres des logiciels de microsimulation permettant de bien reproduire la circulation montréalaise reste limitée : afin d'augmenter la précision des analyses de circulation effectuée sur son territoire, la Direction des transports de la Ville de Montréal a souhaité mener un projet de calibration du logiciel qu'elle utilise (VISSIM). Pour ce projet, la Ville a souhaité étudier les paramètres de comportement routier applicables sur le réseau autoroutier de la métropole puisque, bien que ce réseau ne soit pas de sa responsabilité, les conditions de circulation qui y ont cours affectent directement le réseau artériel de la ville et ne peuvent donc pas être ignorées lors d'une modélisation.

L'objectif est donc d'une part de développer une méthode de calibration systématique des paramètres des logiciels de microsimulation afin que, par la suite, celle-ci puisse être utilisée pour augmenter le nombre de paramètres connus sur le territoire montréalais ou encore être utilisée par d'autres municipalités désireuses de se doter de leur propre jeu de paramètres permettant de décrire la circulation sur leurs territoires respectifs. Le second objectif est d'utiliser cette méthode afin de déterminer les valeurs prises par les paramètres des modèles identifiés pour les besoins de la Ville de Montréal. Les deux objectifs sont intimement liés puisque l'application permet d'identifier les problèmes d'implémentation de la méthode et de proposer les solutions appropriées.

Typiquement, dans les modèles de microsimulation routière, deux modèles se côtoient pour expliquer le comportement des conducteurs : le modèle de poursuite, décrivant comment les véhicules se suivent dans une voie de circulation et le modèle de changement de voie, qui explique

comment un conducteur prend la décision de s'insérer dans la voie adjacente. Dans le cadre de ce travail de maîtrise, le logiciel utilisé est VISSIM et la calibration du modèle Wiedemann 99, utilisé conjointement au modèle de changement de voie, est présentée (PTV, 2013). Du fait de son utilisation en section sans bifurcation (entrée ou sortie), ce modèle constitue un bon choix pour le développement de la procédure en limitant les erreurs possibles de détection en raison des changements de voie.

### **1.3 Approche méthodologique**

La calibration des paramètres nécessite des données de circulations réelles. La première étape du projet de calibration consiste donc à les acquérir. Plusieurs méthodes de collectes existent, mais celle retenue pour cette étude est la collecte de données vidéo, puisque les modèles visés par la calibration demandent l'utilisation de données microscopiques. Pour y arriver, des caméras ont été positionnées sur le réseau autoroutier montréalais en des points représentatifs de différentes infrastructures : sections droites, section avec entrée ou sortie et section d'entrecroisement. Le traitement des images recueillies permet d'obtenir les valeurs requises pour certains paramètres du logiciel comme par exemple les débits et les vitesses. D'autres données dérivées des trajectoires véhiculaires permettent également de comparer les interactions observées et simulées des véhicules. C'est le cas des distributions des temps intervéhiculaires de suivi, soit l'intervalle de temps entre le passage de l'avant d'un véhicule et du nez du véhicule le suivant, qui permet d'évaluer la qualité de la simulation des modèles de poursuite, soit les modèles de Wiedemann dans le cas de VISSIM (PTV, 2013). Pour le modèle de changement de voie, ce sont les distributions des créneaux acceptés, soit le temps intervéhiculaire entre les véhicules présents dans la voie de destination du véhicule effectuant le changement de voie, qui peuvent être utilisées. La calibration des paramètres se fait donc avec une méthode d'optimisation cherchant à reproduire les distributions de ces variables.

La calibration comme telle débute avec le choix d'un algorithme d'optimisation. Dans le cas de cette étude, c'est l'algorithme du *Mesh Adaptive Direct Search* (MADS) qui a été retenu. Les vidéos recueillies permettent de calibrer les modèles de poursuite (Wiedemann 74 et Wiedemann 99) et le modèle de changement de voies. La procédure suivie est globale, puisque tous les modèles compatibles entre eux sont calibrés en même temps, ce qui est rendu nécessaire en raison de leur interaction lors de la simulation.

Finalement, l'étape de validation permet de vérifier que les jeux de paramètres trouvés lors de l'étape de calibration sont valides hors de l'espace-temps où ils ont été étudiés, c'est-à-dire qu'ils ne sont pas restreints aux données utilisées en calibration comme la géométrie des sites de collecte, les débits observés au moment de la collecte, les types de conducteurs et les itinéraires.

## 1.4 Structure du document

Le présent document se divise en six sections. Le premier chapitre permet d'introduire l'importance de la calibration de logiciels de simulation routière et la méthodologie proposée. Suit la revue de la littérature, qui survole les travaux de calibration similaires effectués à travers le monde. Plus spécifiquement, une attention est portée au processus de calibration, aux différents types d'optimisateurs utilisés pour effectuer la calibration, aux questions d'ordre statistique nécessaires durant le traitement des données ainsi qu'au processus de validation des résultats obtenus lors de la calibration.

La description du processus de calibration est effectuée en deux chapitres : le Chapitre 3 présente la méthodologie générale applicable pour n'importe quel modèle alors que le Chapitre 4 présente les portions de la méthodologie qui ont été développées spécifiquement pour le cas d'étude de la calibration du modèle de Wiedemann 99 pour les autoroutes montréalaises. Ces deux chapitres suivent la même structure : y sont présentés tour à tour les phases de préparation, de calibration et de validation. La phase de préparation débute avec la présentation de la méthodologie générale employée. Cette section présente également la formulation du problème, incluant les concepts importants quant aux calculs nécessaires pour les indicateurs de circulation utilisés. Les questions de la collecte et du traitement des données ainsi que de la sélection des paramètres pertinents à la calibration sont également abordées. Les questions d'implémentation du processus d'optimisation permettant d'opérationnaliser la calibration sont présentés dans la section traitant de la phase de calibration. Les sujets abordés sont le choix de la fonction objectif, la formulation des contraintes, le point initial fourni à l'algorithme d'optimisation, le nombre de réPLICATION nécessaires pour assurer une validité des résultats de la simulation et le budget d'évaluations alloué à l'algorithme. La dernière section de ces chapitres présente la validation des résultats, puisque l'implantation de cette phase est très similaire à la phase de calibration.

Finalement, les résultats issus de l'implantation de la méthodologie proposée sur le cas d'étude sont présentés et les sources d'erreurs potentielles sont discutées. Des pistes de solution sont proposées pour améliorer les résultats de futures études utilisant cette technique et des recommandations quant aux valeurs à assigner aux différents paramètres de VISSIM sont formulées. Finalement, des pistes de recherches connexes sont présentées afin d'améliorer les résultats obtenus par l'utilisation de la méthode présentée.

## CHAPITRE 2 REVUE DE LA LITTÉRATURE

Les modèles de microsimulation routière simulent la circulation routière en se basant sur des modèles de poursuite et de changement de voie (FHWA, 2004a). Typiquement, les véhicules sont générés dans un réseau et, à leur entrée, se voient attribuer des caractéristiques (destinations, type de véhicule, comportement, etc.) selon un processus stochastique suivant des distributions statistiques prédefinies. Toujours selon (FHWA, 2004a), leurs positions sont ensuite recalculées à intervalles de temps réguliers et durant lesquels les différentes interactions seront recalculées. La longueur exacte de l'intervalle de temps est souvent un paramètre du logiciel qui peut être contrôlé par l'utilisateur. Chaque modèle de microsimulation présente un jeu de paramètres qui permettent à l'utilisateur de calibrer les modèles pour mieux refléter les conditions locales à l'étude (FHWA, 2004b). L'attribution aléatoire des valeurs individuelles peut dépendre d'un chiffre aléatoire d'amorce (*seed number*) qui détermine la suite de valeurs aléatoires générées (FHWA, 2004b). Modifier ce chiffre a pour conséquence de modifier certaines valeurs attribuées aux véhicules et de modifier donc les résultats de la simulation. À l'opposé, les modèles déterministes n'impliquent aucune variabilité de la simulation une fois les paramètres fixés (FHWA, 2004a). Leur fonctionnement n'implique d'ailleurs pas une simulation du réseau, mais plutôt un calcul basé sur des relations entre des caractéristiques du réseau et des indicateurs de l'état de la circulation. Ces modèles produisent une réponse rapide, mais leur capacité à analyser des réseaux complexes ou à prendre en compte des effets de système est limitée (FHWA, 2004b), puisqu'il n'y a pas de mesure de la variabilité du système réel.

VISSIM répond à la définition ci-haut proposée par la Federal Highway Administration (FHWA). Dans VISSIM, des paramètres clés tels que l'arrivée du véhicule dans le réseau et la vitesse désirée d'un conducteur sont systématiquement générées aléatoirement. Certains autres paramètres peuvent être prédéterminés ou aléatoires. C'est le cas de la décision d'itinéraire des véhicules qui peut être une probabilité à chaque embranchement ou une matrice des paires origine-destination préfixées, au choix de l'utilisateur. Les autres paramètres, fixes, représentent plutôt des conditions de conduite, par exemple le champ de vision du conducteur, et sont constants tout au long de la simulation et d'une réPLICATION à l'autre. La variabilité des paramètres aléatoires affecte les interactions entre les agents simulés, ce qui rend le résultat de la simulation non-prédictible directement à partir des valeurs des différents paramètres.

Dans VISSIM, les comportements routiers sont simulés en utilisant quatre algorithmes principaux, soit un *modèle de succession* (poursuite) des véhicules, un *modèle de changement de voie*, un *modèle de comportement transversal*, soit le modèle décrivant la position d'un véhicule à l'intérieur de sa voie de circulation, et un *modèle de comportement aux installations de signalisation lumineuse* (PTV, 2013). Ces différents modèles reposent chacun sur plusieurs paramètres permettant d'ajuster les comportements routiers aux conditions observées sur un réseau réel. L'utilisation de modèles non calibrés ou mal calibrés peut entraîner la production de résultats erronés.

## 2.1 Principes de calibration des logiciels de microsimulation

La calibration est un processus qui permet d'ajuster les paramètres d'un modèle pour que les résultats qu'il produit correspondent aux conditions réelles de trafic. Ce processus implique d'ajuster un nombre potentiellement très élevé de paramètres. Selon (FHWA, 2004b), il est ainsi essentiel de séparer le processus en plusieurs étapes afin d'éviter une boucle infinie en ajustant continuellement la valeur des paramètres pour régler des problèmes dont la résolution fera apparaître de nouveaux problèmes.

Dans le domaine de l'optimisation, ce processus est connu comme un problème inverse, puisqu'on cherche l'état du modèle permettant d'obtenir une réponse donnée (notamment les conditions observées sur le réseau devant être simulé) plutôt que de rechercher la meilleure réponse pouvant être obtenue à partir du modèle (par exemple le plus haut débit de circulation associé à un modèle de poursuite donnée).

Un cadre général de calibration des modèles de microsimulation se dégage de la littérature : il se déroule en trois phases, soit la *phase de préparation*, la *phase de calibration* et la *phase de validation* (Hellinga, 1998; Hollander & Liu, 2008b; Li, Liu, & Li, 2010; Park & Kwak, 2011; Park & Qi, 2005, 2006; Park & Schneeberger, 2003).

### 2.1.1 Phase de préparation

Cette phase englobe plusieurs étapes importantes, dont l'identification de la portée de l'étude, la collecte de données, la production du réseau virtuel à simuler et le choix des indicateurs (variables à comparer entre la simulation et la réalité) et des fonctions d'ajustement (les mesures statistiques

permettant de comparer les indicateurs mesurés dans la réalité et ceux calculés à partir des résultats d'un modèle) (Hollander & Liu, 2008b; Li et al., 2010). Les auteurs de (Park & Qi, 2005; PTV, 2013) ajoutent à cette liste l'identification des paramètres à calibrer. Cette identification, qui sera traitée plus en détail à la section 2.2, se fera en lien avec l'objectif et la portée de l'étude à effectuer, qui constitue la première étape dans le processus de calibration. On notera tout de suite que les différentes étapes de cette phase sont interdépendantes, et que les choix faits à une étape, par exemple le choix des données collectées, influencent ceux des étapes subséquentes, par exemple le choix des fonctions d'ajustement. C'est pourquoi il peut être opportun de revenir à une étape précédente lorsque les limitations à une des étapes subséquentes ne permettent pas d'atteindre l'objectif. C'est le cas notamment lorsque l'indisponibilité de certaines données ne permet pas d'atteindre une partie des objectifs de la calibration.

L'identification des objectifs et de la portée de la calibration est la pierre d'assise du processus, puisqu'il permet de tracer un plan de la calibration et des besoins à chaque étape. C'est dans cette étape que l'on déterminera le besoin d'utiliser ou non un logiciel de microsimulation pour l'étude à effectuer (Hellinga, 1998) et, le cas échéant, de l'étendue du réseau à simuler (Hollander & Liu, 2008b). Les différents scénarios étudiés ou les différents sous-problèmes impliqués dans l'étude devraient être identifiés, puisque leurs besoins spécifiques en termes de données pourraient différer (Hollander & Liu, 2008b). À cette étape, il pourrait paraître tentant de décomposer un problème complexe en différents sous-problèmes plus simples et moins demandant en puissance de calcul, par exemple en isolant les différents modèles impliqués, puis de les résoudre indépendamment et à la chaîne. Toutefois, comme le soulignent (Hollander & Liu, 2008b), cette approche risque de ne pas donner de bons résultats si les sous-problèmes ne sont pas parfaitement indépendants. Dans le cas contraire, soit dans la majorité des problèmes de calibrations, une stratégie permettant la calibration globale du problème devrait être utilisée plutôt que d'effectuer la division en sous-problèmes.

L'étape suivante est bien souvent l'un des plus grands défis de la phase de préparation : l'acquisition des données. Selon (Hellinga, 1998), le type de données utilisé ainsi que leurs résolutions et leurs couvertures, tant spatiales que temporelles, devraient être énoncées et leur importance devrait être décidée en lien avec le projet et avec les indicateurs choisis. Certaines données de base sont indispensables, telles que les informations géométriques à propos du réseau à reproduire : longueur des liens, nombre de voies, signalisation, etc. Ces données sont cependant

de nos jours facilement accessibles grâce aux données géospatiales, tel que *Google Maps* ou *OpenStreetMap*. Dans le cas d'autres types de données, leur absence est un obstacle difficilement contournable et cette absence est souvent due aux coûts ou à la difficulté d'obtenir de telles données (Antoniou et al., 2014). Il faut alors évaluer s'il est possible de les prendre d'une source externe au projet, par exemple une autre étude menée avec des conditions semblables. Si une telle décision est prise, il est alors impératif de spécifier toutes les hypothèses énoncées permettant l'utilisation de ces données (Antoniou et al., 2014). Dans le cas contraire, il est possible qu'il faille retourner à la définition du projet et en changer la portée et les objectifs pour qu'il devienne possible de tous les réaliser.

Un autre aspect important à considérer lors de la collecte de données est le besoin de reproduire une vaste gamme de conditions de circulation afin d'être en mesure de prédire la circulation et les indicateurs de performance du système prévalant dans les scénarios hypothétiques testés. De même, les données récoltées doivent, au besoin, permettre d'évaluer des changements de comportement hors du contexte pur de la circulation, comme des changements d'itinéraire ou des transferts modaux, qui peuvent émerger à la suite des changements géométriques ou de réglementations testés (comme le changement des limites de vitesses).

Même une fois calibré, un modèle ne garantit aucunement de bien reproduire les conditions de circulation hors de sa plage de calibration. Par conséquent, des données variées permettant de reproduire les conditions prévisibles dans les différents scénarios étudiés devraient être collectées (Hollander & Liu, 2008b). Ceci permet de pouvoir évaluer la précision des prédictions du modèle et de relativiser les résultats obtenus. La diversité des données est également importante afin d'éviter le surajustement (*overfitting*) du modèle à des conditions particulières de circulation lors de la calibration. Le surajustement a comme conséquence une réduction de la capacité du modèle à prévoir adéquatement des conditions différentes de celles prévalant dans la plage de donnée pour lequel il est ajusté. Une formulation judicieuse du problème peut également aider à régler ce problème.

Les indicateurs sont les mesures de la circulation utilisées pour comparer les simulations du modèle à la réalité. Ces mêmes indicateurs et d'autres peuvent ensuite être utilisés pour l'objectif initial du développement du modèle, par exemple pour comparer des alternatives d'aménagement de transport. Des indicateurs souvent utilisés sont le débit, le temps de parcours, la longueur de la file

d'attente et les émissions des véhicules. Selon (Hellinga, 1998), il n'y a pas de limite au nombre d'indicateurs que l'analyste peut utiliser pour décrire la circulation lors de son étude. Ces indicateurs pourront être utilisés dans le processus de calibration ou encore comme contraintes ou encore comme mesure permettant de comparer des scénarios. Ces différentes fonctions peuvent toutes être basées sur les mêmes indicateurs ou encore en utiliser des différents. Un troisième usage possible des indicateurs supplémentaires peut être de départager deux scénarios à priori équivalents, lorsque comparés selon les indicateurs principaux choisis, mais dont l'un présenterait un avantage sur un des indicateurs supplémentaires. Afin d'évaluer la capacité du modèle à reproduire les conditions de circulation réelles, un seul indicateur a été utilisé par (S.-J. K. Kim, Wonho & Rilett, 2005; Li et al., 2010) alors que (Chu, Liu, Oh, & Recker, 2003; Oketch & Carrick, 2005; Otković, Tollazzi, & Šraml, 2013; Toledo et al., 2003) en ont utilisés plus d'un.

Le choix des fonctions d'ajustement est au cœur du processus de calibration puisque c'est ce qui permet de comparer les indicateurs calculés sur les résultats de la simulation et ceux calculés sur les données terrains. Cette comparaison donne une évaluation de l'erreur que commet le modèle en tentant de reproduire un phénomène donné. Les sommes ou moyennes d'erreurs sont utilisées soit sur des éléments différents, dans le cas de valeurs agrégées comme des débits, soit sur des distributions, par exemple la vitesse de tous les véhicules ayant empruntés un lien. La nature et la quantité des données déterminent en partie le choix de la fonction d'ajustement utilisée pour évaluer un indicateur donné et différents indicateurs peuvent être évalués avec des fonctions d'ajustement différentes. Le choix de la fonction influence donc directement le processus de calibration puisque les différentes fonctions ne traitent pas toutes également les données : certaines sont plus sensibles que d'autres aux valeurs extrêmes. Dans le même ordre d'idée, certaines fonction sont plus appropriées pour traiter des distributions alors que d'autres sont utiles pour des indicateurs agrégés comme la moyenne.

Le nombre de fonctions différentes utilisées dans la littérature est grand : pourcentage d'erreur (Park & Qi, 2007; Zhizhou, Jian, & Xiaoguang, 2005), moyenne des erreurs relatives (Toledo et al., 2003), somme des erreurs quadratiques (Yu, Li, & Zhuo, 2004), racine de l'erreur quadratique moyenne (Zhizhou et al., 2005), racine de l'erreur quadratique moyenne normalisée (Toledo et al., 2003), moyenne des erreurs absolues (K.-O. Kim & Rilett, 2003; Li et al., 2010), et plusieurs autres. Par contre, ce ne sont pas toutes les fonctions d'ajustement qui sont appropriées pour toutes les situations. Selon (Antoniou et al., 2014; Hollander & Liu, 2008b), les fonctions telles que le

pourcentage d'erreur, la moyenne des erreurs ou la moyenne des erreurs relatives laissent des erreurs de signes opposés s'annuler et sont donc inappropriées dans le contexte d'une calibration. D'autres indicateurs, comme la moyenne des erreurs absolues et la moyenne relative des erreurs absolues pénalisent toutes les erreurs, quelle que soit leur distance à la moyenne, de la même façon. Or, la calibration d'un modèle stochastique implique que des différences auront lieu entre les modélisations et il est donc normal de retrouver de faibles erreurs autour de la moyenne. Ainsi, les tests pénalisant plus fortement les erreurs éloignées de la moyenne, comme l'erreur quadratique, la racine de l'erreur quadratique moyenne ou encore la racine de l'erreur relative quadratique normalisée sont plus appropriées (Hollander & Liu, 2008b).

Selon (Toledo & Koutsopoulos, 2004), les tests statistiques, tels que le test de Student bivarié, utilisé par (Park & Schneeberger, 2003), le test de Mann-Whitney, le test de Kolmogorov-Smirnov, utilisé par (S.-J. K. Kim, Wonho & Rilett, 2005), et les intervalles de confiances peuvent également être des fonctions d'ajustement intéressantes à utiliser dans le cadre d'un projet de calibration pour des indicateurs sous forme de distributions statistiques, comme par exemples les vitesses des véhicules sur un lien. Ces tests prennent comme hypothèse que les distributions simulées et observées sont indépendantes et évaluent si elles proviennent de la même loi statistique. Le test de Student est paramétrique : il repose sur l'hypothèse que les deux distributions sont normales, ce qui le rend moins flexible, surtout en présence de distributions dont la loi n'est pas forcément connue. Un autre test paramétrique souvent retrouvé dans la littérature est l'analyse de variance (ANOVA), utilisée entre autres par (Hollander & Liu, 2008b; Lownes & Machemehl, 2006; Park & Kwak, 2011), qui teste que deux distributions ou plus, présumées normales et de même variance, suivent à la même loi. Son pendant non paramétrique est le test de Kruskal-Wallis. Par contre, ces deux test ne permettent pas de déterminer directement laquelle ou lesquelles des distributions testées ne suivent pas la même loi que les autres (Ott & Longnecker, 2010), mais peut tester de façon efficace plusieurs distributions en même temps, ce qui peut être utile dans un cas où il n'est pas utile de connaître quelles distributions sont différentes des autres. Si on désire les détecter, on doit alors appliquer des tests successivement à chaque paire de distributions.

Le choix des paramètres à étudier est une étape cruciale de la phase de préparation. En effet, tous les paramètres qu'il est possible de modifier n'ont pas la même importance dans les résultats obtenus. De plus, le coût de calibration augmente avec le nombre de paramètres à optimiser. Un plus petit nombre de paramètres permet à l'analyste de mieux suivre l'impact de chaque

modification sur les résultats alors qu'un grand nombre de paramètres demande l'assistance d'algorithmes automatisés (Hollander & Liu, 2008b). Ces algorithmes ont cependant de meilleures chances d'arriver à un résultat mieux ajusté que la modification manuelle des paramètres. Une des méthodes de réduction du nombre de paramètres est l'élimination. Certains paramètres peuvent être rapidement écartés puisqu'ils n'ont tout simplement aucune chance d'avoir un impact sur les indicateurs utilisés dans l'analyse de la simulation. C'est le cas, par exemple, de tous les paramètres d'un modèle de comportement des piétons dans le cas d'une simulation n'incluant pas de piétons. Si l'analyste n'est pas sûr de l'effet d'un paramètre sur les résultats, il devrait soit être conservé, soit soumis à une analyse de sensibilité (Miller, 2009) (voir section 2.2).

L'espace de définition des paramètres doit être déterminé selon la littérature et le manuel d'utilisation du logiciel utilisé (Oketch & Carrick, 2005). Certains paramètres auront des bornes définies par le logiciel, et leur transgression causera un malfonctionnement du logiciel. D'autres paramètres devront être bornés pour éviter que des situations improbables ou erronées ne se produisent dans la simulation (Miller, 2009). Par exemple, l'analyste voudra éviter que les véhicules se « superposent » ou entrent en collision dans le réseau. Une vérification visuelle du déroulement de la simulation peut d'ailleurs aider à trouver ces problèmes. D'autres paramètres encore auront une signification physique que l'analyste pourra vouloir conserver lors de la calibration, et choisira ainsi des bornes qui conserveront ce sens. Par exemple, il serait étrange d'obtenir une valeur négative pour la variable représentant la distance à l'arrêt entre deux véhicules, même si son utilisation permet au modèle de générer d'excellents résultats. Si tel est le cas, il se peut que le modèle soit lui-même le problème puisqu'il ne permet pas de décrire une situation donnée. Si sa formulation est disponible, il se peut qu'il soit préférable de revoir le modèle pour l'améliorer plutôt que d'investir énormément d'énergie à tenter de le calibrer sans garantie de réellement minimiser l'erreur qu'il génère.

### **2.1.2 Phase de calibration (optimisation)**

La phase de calibration est la phase dédiée à rechercher le meilleur jeu de paramètres, aussi appelées variables d'optimisation dans la littérature mathématique, afin que le modèle reproduise le plus fidèlement possible au travers des indicateurs choisis les observations terrain. Cette phase suit le choix des paramètres à conserver (Antoniou et al., 2014) effectué lors de la phase de préparation et lors de l'analyse de sensibilité, le cas échéant.

La formulation du problème est sa représentation mathématique permettant de conduire la calibration. Cette représentation inclut minimalement une fonction objectif à minimiser et peut également inclure des contraintes. La fonction objectif, dans le contexte d'un problème de calibration, est l'élément qui permet de ne donner qu'une seule valeur à la qualité de la simulation et peut combiner plusieurs fonctions d'ajustement qui évaluent chacune, à l'aide d'un indicateur, l'erreur que commet le modèle à reproduire les conditions observées en réalité. C'est ce qui permet à l'algorithme d'optimisation d'évaluer si ses essais successifs améliorent ou détériorent la simulation et de déterminer le prochain jeu de paramètres à tester. Des exemples de fonction objectif sont des fonctions de poids, permettant de donner un impact différents aux fonctions d'ajustement utilisées ou encore le maximum des résultats des fonctions d'ajustement. Il faut faire attention aux échelles des unités des différentes fonctions d'ajustement utilisées pour éviter que des variations significatives d'une des fonctions puissent être complètement masquées par de faibles variations d'une autre fonction d'ajustement. Au besoin, il faudra normaliser les échelles. Les contraintes de calibration sont des conditions que l'algorithme d'optimisation doit respecter. Elles peuvent être formulées à partir de la valeur de certains paramètres, à partir d'indicateurs ou de fonctions d'ajustements. Elles servent à donner un cadre à la simulation qui ne peut pas être traduit dans la fonction d'ajustement. Par exemple, une contrainte pourrait être formulée pour vérifier que les véhicules simulés n'ont pas un comportement anormal ou encore qu'un certain nombre de véhicules soit affectés à emprunter un chemin donné dans le réseau si un modèle d'affectation des itinéraires est utilisée.

La formulation du problème est également le moment de sélectionner l'algorithme d'optimisation qui sera utilisé pour effectuer la calibration. Cette question constitue le cœur du travail de calibration et plusieurs considérations doivent être analysées. Premièrement, il doit permettre l'évaluation du problème tel que défini. Il doit également ne pas être trop coûteux en termes de calculs afin de permettre à l'analyste d'effectuer la calibration en un laps de temps raisonnable. Les auteurs de (Hollander & Liu, 2008b) ont relevé dans une revue de 18 études utilisant différents logiciels de microsimulation que les algorithmes utilisés ont tous en commun de ne pas utiliser les dérivés des problèmes à calibrer puisque leurs formulations explicites, incluant la simulation des résultats via le logiciel, sont trop complexes, inconnues ou non dérивables, comme c'est le cas lors de l'utilisation de fonction maximum ou minimum. Les algorithmes les plus utilisés sont les algorithmes génétiques, la méthode du Simplex de Nelder-Mead et le *Box's Complex Algorithm*

(Hollander & Liu, 2008b). La section 2.3 traite plus spécifiquement du choix de l'algorithme d'optimisation.

Les logiciels de microsimulation se basant sur une représentation stochastique de la circulation, la question du nombre de simulations nécessaires pour obtenir un résultat statistiquement représentatif devient indispensable à régler. Selon (Antoniou et al., 2014), peu de travaux ont été menés afin de déterminer adéquatement ce nombre. Certaines études utilisent un nombre arbitraire de simulations, c'est le cas de (Park & Qi, 2005, 2006) qui utilise 5 simulations et de (Toledo & Koutsopoulos, 2004) qui en utilise 10 sans justification, alors que d'autres (Jie, Zuylen, Chen, Viti, & Wilmink, 2013; S.-J. K. Kim, Wonho & Rilett, 2005; Toledo et al., 2003; Yu et al., 2004; Zhizhou et al., 2005) ne mentionnent pas le nombre de simulations effectuées. De leur côté, les auteurs de (Chu et al., 2003; Hollander & Liu, 2008b; Li et al., 2010) proposent d'utiliser la formule classique de calcul de la taille d'un échantillon suivante afin de trouver le nombre de simulations nécessaires :

$$N = \left( t_{\alpha/2;N-1} \frac{s}{\bar{x}\varepsilon} \right)^2$$

où  $t_{\alpha/2;N-1}$  est la valeur d'une variable aléatoire  $t$  suivant la loi de Student avec  $N - 1$  degrées de liberté pour un niveau de confiance  $\alpha$  pour  $P(|t| > t_{\alpha/2;N-1}) = \alpha$ ;  $s$  est l'écart-type de l'échantillon,  $\bar{x}$  la moyenne de l'échantillon et  $\varepsilon$  est la tolérance maximale, exprimée en une fraction, sur l'évaluation de la moyenne. L'intervalle de confiance résultant autour de la moyenne est donc  $CI = [\bar{x}(1 - \varepsilon), \bar{x}(1 + \varepsilon)]$ . (FHWA, 2004a; ODOT, 2011) expliquent la procédure itérative pour l'application de cette formule si l'écart-type et la moyenne ne sont pas à priori connus : après avoir déterminé une tolérance jugée acceptable pour l'étude en cours, 10 répétitions sont effectuées donnant l'écart-type de cet échantillon. De la formule, le nombre de répétitions  $N$  requis est calculé. Si  $N$  est plus grand que le nombre de répétitions déjà effectuées, le processus est itéré jusqu'à convergence : on augmente le nombre de répétitions effectuées, les paramètres  $s$  et  $\bar{x}$  sont mis à jour et  $N$  est recalculé. Cette boucle continue jusqu'à ce que  $N$  soit plus petit ou égal au nombre de répétitions effectuées, et ce nombre devient le nombre de répétitions à simuler. Il est à noter que si plus d'un indicateur est utilisé pour décrire la qualité de la simulation,  $N$  doit être calculé pour chacun des indicateurs et la plus grande valeur obtenue doit être retenue.

Les auteurs de (Hollander & Liu, 2008b) soulèvent toutefois que cette méthode pose problème lorsque les fonctions d'ajustement ne se basent pas sur la valeur moyenne d'une variable tirée des données, puisque la formule ne garantit un niveau de confiance que sur une moyenne. (Hollander & Liu, 2008a) propose une méthode alternative où un grand nombre de répétitions est effectué. Le but est de détecter après combien de répétitions la valeur d'un indicateur donné converge vers une valeur stable. La procédure peut être effectuée en prenant plusieurs indicateurs en compte, en sélectionnant le plus grand nombre de répétitions nécessaire pour que tous les indicateurs convergent.

Finalement, la question du temps de « réchauffement » du modèle doit être réglée. Selon (Antoniou et al., 2014), le mode de fonctionnement des logiciels de microsimulation fait en sorte qu'au début de la modélisation, le réseau est vide. Ainsi, lorsque les premiers véhicules sont générés, ils ne rencontrent aucune friction avec d'autres véhicules et leur temps de déplacement est donc grandement réduit. De plus, les différentes interactions qu'on voudrait pouvoir étudier seront faussées par le fait qu'aucun autre véhicule n'interagit avec ces véhicules. Il faut donc laisser le temps au modèle de se retrouver dans un état d'équilibre avant de mesurer les indicateurs. Le temps nécessaire pour atteindre un équilibre dépendra de la longueur du réseau, de sa complexité, de la présence ou non de carrefours à feux, de la vitesse de déplacement, du débit désiré, etc. Trouver le temps exact n'est pas une tâche facile et peut demander plusieurs essais-erreurs (Antoniou et al., 2014). Les auteurs de (PTV, 2013) proposent d'utiliser le temps de déplacement de la plus grande distance du réseau en conditions d'écoulement libre. De son côté, la FHWA (FHWA, 2004a) préconise d'utiliser le double de ce temps ou d'utiliser une méthode graphique permettant de visualiser la fin de la période de réchauffement du modèle.

### **2.1.3 Phase de validation**

La dernière phase est celle de la validation, où on vérifie les résultats de la calibration sur un jeu de données indépendant. Le but est de vérifier si les valeurs obtenues s'appliquent dans une multitude de conditions et non uniquement pour la ou les situations spécifiques étudiées lors de la calibration. Pour cette étape, les mêmes indicateurs et les mêmes fonctions d'ajustement que pour la calibration peuvent être utilisés (Hollander & Liu, 2008b). Le jeu de données peut provenir d'un autre site que celui calibré ou bien d'un autre moment au même site (Li et al., 2010; Park & Kwak, 2011; Park & Qi, 2005), mais ne doit en aucun cas avoir été inclus dans le processus de calibration.

Pour vérifier si le modèle calibré peut être utilisé dans des conditions de circulation autres que celles employées pour la calibration, il peut être validé sur un jeu de données reproduisant ces autres conditions. Dans tous les cas, un modèle ne pourra être utilisé avec un certain niveau de confiance que sur les conditions de circulation ayant été utilisées dans les phases de calibration ou de validation (Hollander & Liu, 2008b). Toute condition extérieure à celles-ci est sujette aux incertitudes liées à l'extrapolation.

Il est souligné dans (Park & Qi, 2005) que souvent, on considère qu'un meilleur niveau de calibration a été obtenu si en plus de l'indicateur utilisé en calibration, par exemple le débit, un autre indicateur complémentaire, par exemple la vitesse, obtient aussi de bons résultats. Les auteurs soulignent toutefois, qu'en pratique, il n'est pas certain que les modèles du logiciel de microsimulation puissent reproduire le trafic avec suffisamment de précision pour que des indicateurs n'ayant pas été inclus dans la fonction objectif ou dans les contraintes aient également été bien calibrés (Park & Qi, 2005). Dans tous les cas, si un modèle est calibré et validé avec certains indicateurs dans la ou les fonctions d'ajustement, alors on ne pourra utiliser avec confiance les paramètres ainsi trouvés que pour reproduire ces indicateurs (Park & Qi, 2005).

#### **2.1.4 Conclusion**

Les étapes discutées ci-haut sont reprises dans le Tableau 2-1, dont les étapes sont largement inspirées de ce qui est présenté dans le résumé de (Hollander & Liu, 2008b). Le lecteur notera toutefois que les différentes étapes listées dans la phase de préparation dépendent les unes des autres et que les choix faits à une des étapes ou des limitations quant à la résolution d'une étape subséquente peuvent entrer en conflit. Il est donc possible que la résolution d'une des étapes amène à revisiter et modifier les choix faits aux autres étapes. Par exemple, l'impossibilité d'accéder à un certain type de données pourrait avoir comme conséquence l'impossibilité de mesurer un indicateur jugé essentiel à la réussite du projet visé, et il sera alors impératif de revoir l'objectif du problème ainsi que les attentes face au processus de calibration avant de poursuivre. Cet exemple illustre également le besoin de planifier les différentes étapes et de vérifier leur faisabilité avant d'engager les ressources nécessaires à la réalisation des étapes plus coûteuses, soit monétairement ou en temps, telle que la collecte de données proprement dite, la construction du modèle dans le logiciel de simulation ou l'analyse de sensibilité.

Tableau 2-1 : Résumé du processus de calibration

Phase	Étapes
Préparation	<ol style="list-style-type: none"> <li>1. Déterminer l'objectif du problème</li> <li>2. Déterminer les paramètres pertinents à la calibration et leurs bornes, soit à l'aide d'une analyse de sensibilité, d'une revue de la littérature ou d'études passées pertinentes</li> <li>3. Collecte et traitement des données</li> <li>4. Déterminer quels indicateurs (débits, temps de parcours, etc.) sont utilisés pour décrire l'état de la circulation, tant dans les simulations que dans les données terrains</li> <li>5. Déterminer les fonctions d'ajustement utilisées pour comparer les indicateurs simulés et les indicateurs terrain</li> </ol>
Calibration	<ol style="list-style-type: none"> <li>6. Formuler la fonction objectif</li> <li>7. Formuler les contraintes de calibration</li> <li>8. Déterminer le nombre de répétitions de la simulation à effectuer</li> </ol>
Validation	<ol style="list-style-type: none"> <li>9. Calcul de la fonction objectif sur un jeu indépendant de données avec les paramètres calibrés</li> </ol>

## 2.2 Analyse de sensibilité

Dans le contexte de la calibration d'un modèle de microsimulation, le but d'une analyse de sensibilité est de déterminer quels paramètres influencent le plus les résultats de l'analyse projetée (Yu et al., 2004). En connaissant ces influences, il est possible de concentrer les ressources de calcul sur un sous-ensemble de paramètres.

L'analyse de sensibilité peut prendre plusieurs formes. Sa version la plus simple, parfois appelée *One Factor at a Time*, consiste à tester successivement plusieurs valeurs d'un paramètre sans modifier les autres afin de voir l'effet de cette variation sur l'indicateur choisi. Cette méthode, simple et intuitive, peut cependant masquer les effets d'interaction entre les paramètres : certains paramètres n'ayant à priori aucun effet se révèleront très influents lorsque modifiés en conjonction avec un autre paramètre (Antoniou et al., 2014). De plus, la plage de validité d'une telle analyse se

limite souvent à quelques points autour des valeurs par défaut et il n'est pas clair que les résultats peuvent être extrapolés sur le domaine entier.

Une version légèrement plus complète peut être trouvée dans la littérature sur les logiciels de microsimulation. C'est le cas de (Lownes & Machemehl, 2006) qui fait une analyse de sensibilité groupée, c'est-à-dire que certains paramètres sont variés conjointement alors que les autres sont laissés fixes. Les groupements ont été créés à partir de ce qu'il semblait logique de grouper pour le chercheur en se référant aux définitions données par le fabricant de VISSIM des différents paramètres du modèle de Wiedemann 99. Cette approche ne teste cependant pas toutes les combinaisons, et peut par conséquent encore masquer certains effets.

Selon (Saltelli, Tarantola, Campolongo, & Ratto, 2004), une bonne analyse de sensibilité doit répondre aux critères suivants :

1. Elle doit être capable de prendre en compte des paramètres évoluant sur des échelles de grandeur différentes et qui sont générés en tirant des valeurs dans des distributions statistiques qui peuvent être différentes.
2. Elle doit inclure une analyse multidimensionnelle, c'est-à-dire être capable d'évaluer l'effet d'un paramètre conjointement avec d'autres paramètres.
3. Elle doit être indépendante du modèle étudié, c'est-à-dire qu'elle doit pouvoir donner des résultats fiables peu importe la nature du modèle et être capable de capturer les effets d'interaction entre les paramètres.

Afin de répondre à ces critères et de remplir l'objectif d'identifier les paramètres les plus influents, (Miller, 2009; Saltelli et al., 2004) proposent l'utilisation de la méthode de Monte Carlo. Les auteurs de (Antoniou et al., 2014) listent plutôt l'ensemble des algorithmes possibles pour l'analyse de sensibilité, dont la méthode de Monte Carlo. Cette méthode consiste à générer des valeurs aléatoires pour chacun des paramètres, selon une distribution uniforme entre leurs bornes, et d'effectuer la simulation pour ces valeurs. Pour que la méthode fonctionne, le nombre de points (ensembles de valeurs des paramètres) ainsi générés doit se situer entre 100 et 1000. Un test statistique permet ensuite de séparer les paramètres en deux groupes : ceux ayant un impact significatif et ceux n'ayant pas un impact significatif sur le ou les indicateurs d'intérêt.

## 2.3 Les algorithmes d'optimisation

La littérature en calibration de logiciel de microsimulation contient plusieurs algorithmes différents, allant des algorithmes génétiques (*Genetic Algorithm*) (S.-J. K. Kim, Wonho & Rilett, 2005; Li et al., 2010; Park & Qi, 2006, 2007; Yu et al., 2004; Zhizhou et al., 2005), aux essaims de particules (*Particule Swarm*) (Aghabayk, Sarvi, Young, & Kautzsch, 2013) et à l'algorithme du simplex de Nelder-Mead (K.-O. Kim & Rilett, 2003). (Ciuffo, Punzo, & Montanino, 2012) étudient encore d'autres algorithmes tel que le *Simulated Annealing* et le *OptQuest Multistart*. D'autres études encore utilisent plutôt des techniques de discréétisation et d'échantillonnage de l'espace de recherche telles que l'échantillonnage par hypercubes latin (*Latin Hypercube Sampling*) (Park & Schneeberger, 2003) ou la recherche exhaustive (*Exhaustive search method*) (Chatterjee, Edara, Menneni, & Sun, 2009), mais qui ne sont pas des algorithmes d'optimisation à proprement parler.

À l'exception du simplexe, les algorithmes présentés ci-haut sont des heuristiques visant à organiser une méthode de recherche essentiellement basée sur des points générés aléatoirement afin de faire progresser la réponse vers une solution qu'on espère optimale. L'algorithme génétique utilise des principes empruntés à la théorie de l'évolution, avec la création d'une population de points, la sélection des meilleurs candidats, leur reproduction et des mutations apportées à la prochaine génération (Eiben, Raué, & Ruttkay, 1994). L'essaim de particules tend à recréer un essaim d'abeilles artificielles (particules) en générant un nuage de points, étant chacun un ensemble de valeurs pour chacun des paramètres, et en orientant son déplacement vers les solutions les plus prometteuses trouvées à l'itération courante de l'algorithme (Eberhart & Kennedy, 1995). Toutes ces méthodes ont pour point commun de nécessiter un très grand nombre d'évaluations pour bien fonctionner, et ce à chaque itération. Le coût en terme de temps de calcul devient donc prohibitif, d'autant plus que dans leur analyse de l'algorithme génétique, les auteurs de (Ciuffo et al., 2012) concluent que la capacité d'un tel algorithme à trouver un optimum local est souvent moindre que celle de plusieurs autres algorithmes.

L'échantillonnage par hypercubes latins est une méthode statistique servant à générer des ensembles de paramètres couvrant bien l'espace de définition. Pour y arriver, l'algorithme découpe l'espace de calcul en sous-espaces, en divisant chaque dimension en autant de régions que de points doivent être sélectionnées. En deux dimensions, cela peut être visualisé par une matrice  $p \times p$  où  $p$  est égal au nombre de points à générer. L'algorithme tire ensuite un point dans une case de cette

matrice et bloque ensuite la ligne et la colonne correspondante lors du tirage du prochain point. Cette méthode d'échantillonnage assure une meilleure couverture que ce qu'un échantillonnage aléatoire peut normalement générer en réduisant la corrélation entre les dimensions des paramètres et en maximisant la couverture de l'espace (Saltelli et al., 2004). L'échantillonnage peut efficacement être couplé à un algorithme utilisant une stratégie d'exploration aléatoire de l'espace afin de fournir des candidats, mais elle ne constitue pas un algorithme d'optimisation en soi puisqu'aucun choix n'est fait en fonction des résultats obtenus.

Appliqué à la calibration, le réseau de neurones tente de constituer une évaluation de la fonction à calibrer (Otković et al., 2013) et de la substituer à l'utilisation du logiciel afin d'accélérer le processus de calibration. En ce sens, pour ce qui est de la calibration de logiciels de microsimulation, il n'est pas utilisé comme un algorithme d'optimisation mais plutôt comme une stratégie se combinant avec un autre algorithme.

La méthode du simplexe est quant à elle une des méthodes de recherche directe, également connue sous le nom d'algorithme de Nelder-Mead (Nelder & Mead, 1965). Son fonctionnement consiste à déplacer une forme géométrique composée de  $n + 1$  points dans un univers de dimension  $n$ , correspondant au nombre de paramètres à calibrer (Conn, Scheinberg, & Vincente, 2009). L'algorithme progresse vers une solution optimale en déplaçant itérativement le point ayant la pire solution en effectuant des réflexions, des expansions, des contractions ou des rétrécissements (Kolda, Lewis, & Torczon, 2003). (McKinnon, 1996) démontre cependant que l'algorithme échoue à converger pour certaines familles de fonctions même si celle-ci sont convexes et dérivables. De plus, (Ciuffo et al., 2012) ont testé son applicabilité pour la calibration de logiciel de microsimulation et l'algorithme a échoué tous les tests. Cependant, (K.-O. Kim & Rilett, 2003) ne rapportent pas de problème dans la calibration des modèles d'affectation des logiciels TRANSIM et CORSIM en comparant les débits affectés sur les liens à partir de matrices origines-destinations à des comptages observés sur le réseau.

Au niveau de la littérature en optimisation sans dérivées, on retrouve des algorithmes plus avancés qui découlent des premiers algorithmes de recherche directe. (Audet, 2014) relate les différentes évolutions ayant mené aux algorithmes de recherche directe, dont le *Generating Set Search* (GSS) ou le *Mesh Adaptive Direct Search* (MADS). La particularité de ces algorithmes est de progresser dans l'espace de recherche un point à la fois en construisant une discrétisation de l'espace et en

recherchant les directions de progression les plus prometteuses. Ces algorithmes fonctionnent en deux phases, soit une phase de recherche globale, la *search*, et une phase de sonde, la *poll* qui cherche à cheminer dans l'espace en utilisant les voisins sur la discréétisation de l'espace. La première phase constitue un compromis entre le besoin de progresser dans l'espace vers la solution optimale et celui d'explorer pour une région contenant des meilleurs candidats. Cette phase permet également de sortir des optimums locaux afin de continuer la progression en recherchant plus loin que le voisinage immédiat. Une multitude de stratégies peuvent être utilisées pour générer les points testés durant cette phase.

La discréétisation joue un double rôle en permettant à la fois la génération des candidats à tester pour la phase de *sonde*, mais également en permettant une formulation de la progression de l'algorithme lui conférant une preuve de convergence globale, c'est-à-dire que l'algorithme convergera, vers un optimum local peu importe le point à partir duquel il est initialisé (Conn et al., 2009). Cette convergence est obtenue en réduisant la discréétisation chaque fois qu'aucun des candidats envisagés n'améliore la solution courante. À un optimum, des échecs répétés conduiront éventuellement la taille de la discréétisation à converger vers 0.

Les évolutions les plus récentes de cette famille d'algorithme permettent de minimiser le nombre requis d'évaluations afin de cheminer vers un optimum (Audet, Ianni, Le Digabel, & Tribes, 2014). Ceci contraste d'ailleurs avec le grand nombre d'évaluations requises par les heuristiques évolutionnaires. De plus, la preuve de convergence est intéressante puisqu'elle nous assure, sous certaines conditions, d'obtenir une solution localement optimale.

Le choix de l'algorithme approprié pour un problème donné est une question complexe en soit. Une des préoccupations les plus importantes est la capacité de l'algorithme à fournir une réponse qui soit un optimum local afin d'assurer que le temps de calibration n'ait pas été investi en vain. De plus, le nombre de simulations nécessaire pour que l'algorithme chemine jusqu'à un optimum peut rapidement devenir un facteur critique puisqu'il n'est pas rare que le temps de calcul requis pour effectuer les différentes répétitions soit important. Ensuite, il importe de choisir en fonction de la facilité d'utilisation et du comportement de l'algorithme sur le problème à calibrer, ce qui ne peut pas toujours être connu à priori. Par exemple, les algorithmes génétiques sont simples à planter, mais comportent un très grand nombre de paramètres qu'il faut eux-mêmes calibrer pour chaque problème. S'il est bien calibré, cet algorithme peut avoir de très bonnes performances, mais

le calibrer revient presque à la question de recherche initiale. Finalement, écrire soi-même l'algorithme peut également rendre la tâche globale plus complexe, puisqu'il faudra alors détecter si les erreurs observées proviennent du problème à calibrer ou de l'algorithme. En ce sens, il est plus judicieux d'utiliser un des nombreux algorithmes d'optimisation déjà écrits et disponibles en ligne ou les bibliothèques d'optimisation disponibles pour plusieurs langages de programmation. Ces algorithmes confèrent l'assurance d'avoir été préalablement testés avec succès sur d'autres problèmes et certains ont fait l'objet de multiples publications, augmentant leur crédibilité.

## CHAPITRE 3 MÉTHODOLOGIE GÉNÉRALE

Ce chapitre présente la méthodologie générale de calibration des logiciels de microsimulation développée. Afin de ne pas dédoubler indûment les explications, ce chapitre est succinct et général et plusieurs des questions d'implantations seront abordées lors de la présentation de l'application de la méthode sur la calibration des comportements de conduite dans VISSIM pour les autoroutes montréalaises dans la suite de ce chapitre et dans le chapitre suivant.

La procédure se veut flexible et applicable à n'importe quel type de données, et sur n'importe quel logiciel de microsimulation.

### 3.1 Procédure globale de calibration

La calibration est un cas particulier de l'optimisation, où nous cherchons à minimiser la différence entre les indicateurs observées et les indicateurs simulées. Ce type de problème est connu sous le nom de problème d'optimisation inverse. Mathématiquement, nous cherchons à résoudre le problème suivant pour  $n$  paramètres et  $m$  contraintes :

$$\begin{aligned} & \min_{x \in \Omega} f(x) \\ \Omega = & \left\{ x \in X \subseteq \mathbb{R}^n : c_j(x) \leq 0, j \in \{1, 2, \dots, m\} \right\} \end{aligned}$$

où :

$f$  est la fonction objectif permettant d'évaluer la distance entre les résultats simulés et observés,  $\Omega$  est l'ensemble des points de l'espace réalisable,  $\mathbb{R}^n$  est l'espace de recherche dont les  $n$  dimensions correspondent aux  $n$  paramètres inclus dans la calibration,  $X$  est le sous-espace formé par les bornes des  $n$  paramètres,  $x$  est un point d'essai, soit un vecteur constitué d'une valeur pour chacun des  $n$  paramètres,  $c_j(x) \leq 0$  est une contrainte de calibration pour  $j \in \{1, 2, \dots, m\}$ .

Les domaines de l'optimisation et du génie civil utilisent des termes différents pour certains des éléments de la calibration, et il est donc nécessaire de fournir une clarification des termes utilisés. Premièrement, les éléments du logiciel de microsimulation pouvant être modifiés et faisant l'objet de la calibration sont nommés les « paramètres du logiciels » abrégés en « paramètres » tout au long du texte, et ils correspondent à ce qui est nommé « variables d'optimisation » dans la littérature mathématique. Pour ce qui est des algorithmes d'optimisation ayant également des

paramètres réglables afin de tenter de trouver une meilleure configuration pour la résolution du problème d'optimisation (par exemple, la taille de la population générée par un algorithme génétique), ceux-ci sont appelés « indicateurs variables » ou « options » dans le présent texte afin d'éviter toute confusion avec les paramètres du logiciel de microsimulation.

### 3.1.1 Schéma

Le schéma général de la phase de calibration est présenté dans la Figure 3-1. Il est composé de plusieurs éléments dont le contenu doit être déterminé durant la phase de préparation ou au tout début de la phase de calibration: les données, la fonction objectif, les contraintes, les paramètres à calibrer et la codification du réseau.

Le logiciel de microsimulation est traité comme si aucune information sur les calculs qu'il effectue n'était disponible. Ces informations sont plus ou moins disponibles d'un logiciel à l'autre, mais cela n'affecte pas la manière de traiter le logiciel puisque même lorsque toutes les équations sont fournies, le système simulé est trop complexe pour utiliser des techniques d'optimisation avec dérivées. Les algorithmes d'optimisation traitent donc le logiciel comme une boîte noire.

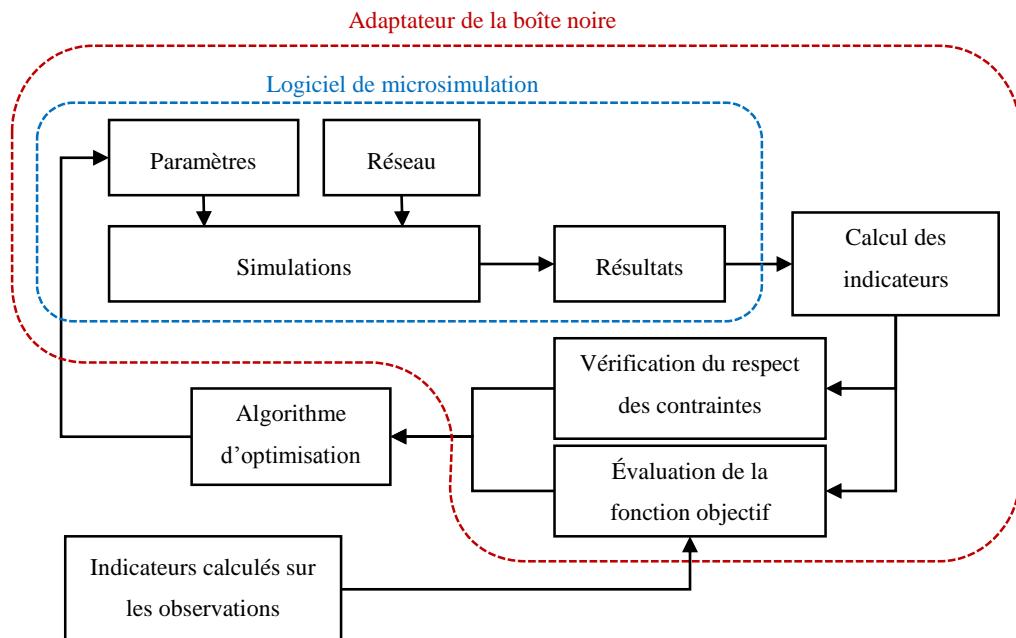


Figure 3-1 : Schéma de la méthodologie générale

Le schéma représente une itération effectuée par l'algorithme d'optimisation choisi pour effectuer la calibration. Celui sélectionne un ou des jeux de paramètres à tester selon la stratégie d'optimisation qui lui est propre. Ces jeux de paramètres sont communiqués au logiciel de microsimulation en passant par un adaptateur permettant d'interpréter les demandes de l'algorithme en des commandes compréhensibles par le logiciel. Le logiciel est ainsi initialisé et les répétitions sont effectuées. L'adaptateur reprend ensuite les sorties du logiciel de microsimulation (les résultats de la simulation) et les transforme, le cas échéant, en des indicateurs pouvant être comparées aux indicateurs mesurés sur le terrain. Il transforme ensuite cette information en un résultat de la fonction objectif, en comparant les données simulées aux données réelles, et des contraintes. Le tout est retourné à l'algorithme d'optimisation qui prend la décision du prochain jeu de paramètres à tester.

Il est important de noter que cette stratégie prend en compte un réseau déjà bien codifié. En effet, bien que la stratégie fonctionne même sur un réseau comportant des erreurs, ce serait une perte de temps que de tenter de calibrer un modèle alors qu'une portion de l'erreur observée est potentiellement due à ces erreurs de codification. Évidemment, pour un réseau de grande taille, il peut être difficile d'éliminer toutes les erreurs de codification et nous définissons donc un réseau bien codifié comme étant un réseau permettant de simuler de façon satisfaisante, pour les besoins du problème à l'étude, le réseau réel. Si une erreur majeure de codification est par la suite découverte dans ce réseau, il faudra reprendre le processus de calibration.

En cas de parallélisation des calculs le schéma ci-haut est appelé pour chaque jeu de paramètres devant être testés. Deux stratégies sont possibles : soit l'algorithme d'optimisation gère lui-même la parallélisation et le nombre de points à tester simultanément en appelant lui-même la méthodologie complète pour chacun des points, soit l'adaptateur contient une stratégie de parallélisation. Cette deuxième approche a été retenue dans l'implémentation présentée plus loin dans ce mémoire, puisque la majorité des licences des logiciels de microsimulation limitent le nombre d'instances pouvant travailler simultanément. Il devient alors plus pratique que l'adaptateur gère directement la stratégie de parallélisation, par exemple en parallélisant différemment les simulations et le traitement de leurs résultats, afin de tirer le maximum du nombre d'instances du logiciel disponibles.

La phase de validation reprend ce schéma, avec la différence que l'algorithme de calibration n'est pas présent et que la boucle de rétroaction est ainsi coupée. En effet, seul un essai est fait lors de la validation et en l'absence de résultats satisfaisant ou devant des contraintes non respectées, le processus de calibration en entier doit être repris ou l'historique de calibration doit être investigué afin de trouver un point réalisable préalablement trouvé.

### **3.1.2 Un logiciel permettant l'automatisation du processus : CATS**

Afin d'automatiser le processus de calibration, un logiciel, écrit dans le langage Python, a été écrit et utilisé. Nommé CATS (*CAlibration of Traffic Simulations*), il implémente pour l'instant le traitement des données, des indicateurs et des fonctions d'ajustement utiles pour l'analyse présentée à la section 3.1.1 pour le logiciel VISSIM et pour un seul algorithme d'optimisation. Cet outil pourra être généralisé, en lui permettant de gérer d'autres types d'indicateurs, de données et d'autres logiciels de microsimulation. Cette dernière fonctionnalité peut cependant être longue à planter puisque les modèles et leurs degrés de liberté, de même que leur possibilité de communication via une interface de programmation différent d'un logiciel à l'autre. Toutefois, l'extension de CATS à d'autres indicateurs, d'autres types de données et d'autres fonctions d'ajustement ne requiert pas beaucoup de travail et, évidemment, la portion la plus complexe de la calibration demeure la définition des besoins pour une étude donnée en fonction des objectifs.

Cet outil permet, outre la calibration des logiciels de microsimulation, d'effectuer des analyses de sensibilité et de la validation. Il inclut également quelques scripts permettant une visualisation de certains aspects, tels que l'historique de calibration, permettant une prise de décision plus efficace entre les essais de calibration. Ses différentes composantes sont décrites en annexe et le code est disponible gratuitement à l'adresse suivante : <https://bitbucket.org/LaurentGauthier/cats>.

## **3.2 Phase de préparation**

### **3.2.1 Formulation du problème**

#### **3.2.1.1 Les indicateurs utilisés**

Afin de mener à bien le processus de calibration, le logiciel de microsimulation est traité comme une boîte noire. Un algorithme d'optimisation sans dérivées, choisi à la section 3.2.2, aura comme

tâche de trouver le jeu de paramètres permettant de mieux représenter la circulation routière. Du point de vue de cet algorithme, tous les traitements effectués sur les résultats de la simulation et décrit dans le choix des indicateurs, de la fonction objectif et de la formulation des contraintes font partie intégrante de la boîte noire.

La calibration des différents modèles se fait grâce à la comparaison entre les temps intervéhiculaires (TIV) observés et ceux du modèle de simulation. Ce choix d'indicateur s'établit naturellement puisque le phénomène décrit par les modèles de poursuites et de changement de voie sont caractérisés par les temps intervéhiculaires. Pour VISSIM, le modèle de poursuite est intrinsèquement basé sur la distance et le différentiel de vitesse entre les véhicules et les TIV peuvent être estimés à partir de ces deux variables. Il est donc possible de caractériser le modèle à l'aide des TIV. De plus cet indicateur est beaucoup plus facile à extraire des données vidéo que la distance et la vitesse, qui sont sujettes à plus d'erreurs. Pour les TIV simulés, ils sont facilement dérивables à partir du fichier de trajectoire généré par le logiciel à la fin de chacune des simulations et qui rapportent les positions successives des véhicules dans le réseau simulé, donnant un enregistrement par pas de simulation. Pour le modèle de changement de voies, le logiciel se base explicitement sur les créneaux acceptés (PTV, 2013). Pour calibrer le modèle de poursuite, les TIV utilisés sont directement observés entre deux véhicules se suivant dans une même voie de circulation. Pour calibrer le modèle de changement de voie, ce sont plutôt les TIV acceptés qui seront utilisés, soit le TIV de suivi dans la voie de destination immédiatement avant le changement de voie.

Le point exact de calcul des TIV de suivi ne semble pas important en général. Toutefois, les segments directement devant les entrées et sorties sont moins propices, puisque plus de mouvements d'entrecroisement auront lieu à ces endroits. Dans le cas des vidéos, un processus par essais et erreurs a permis de déterminer qu'il est préférable d'effectuer ce calcul lorsque les véhicules sont assez proches de la caméra. En effet, l'erreur de parallaxe est ainsi moindre que si le calcul était effectué avec des véhicules situés loin dans le champ de vision de la caméra. Ainsi, le calcul est effectué à 20 % de la distance maximale d'analyse vidéo par rapport à la caméra, ce qui peut être visualisé à la Figure 3-3. Cette distance a été obtenue en effectuant plusieurs essais et en comparant le taux de détections des TIV avec le nombre total de véhicules ayant été détecté sur le lien, certains véhicules mettant plus de temps à être détectés après leur entrée dans le champ de vision de la caméra que d'autres. Le TIV de changement de voie est quant à lui calculé à l'endroit

où le véhicule qui change de voie arrive dans la voie de destination (soit  $x_2$  dans la Figure 3-2). La méthode de calcul est la même pour les changements de voies observés dans les données vidéos et pour ceux de la simulation.

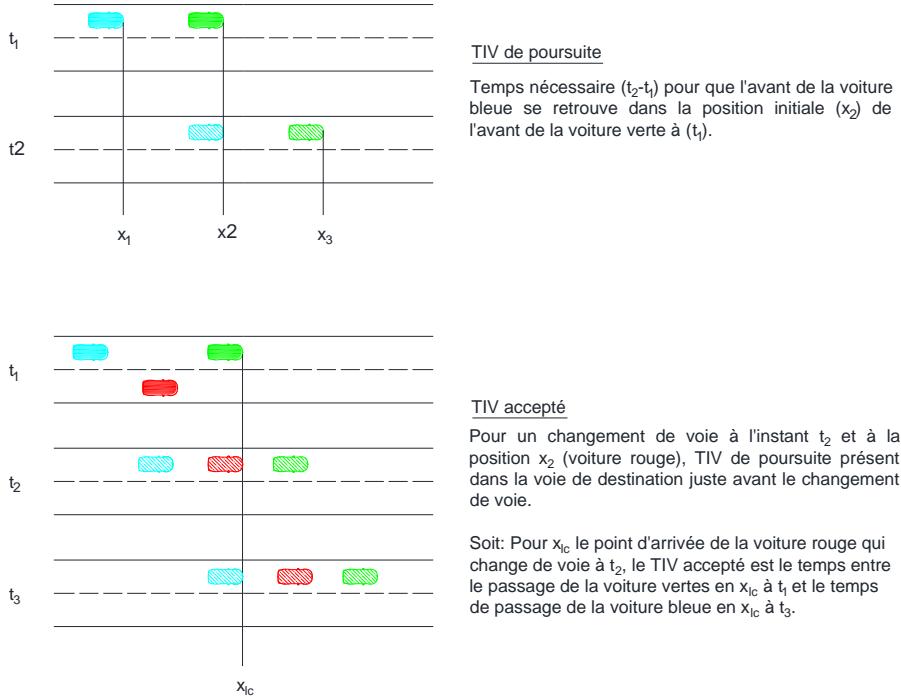


Figure 3-2 : Illustration des différents types de temps intervéhiculaires

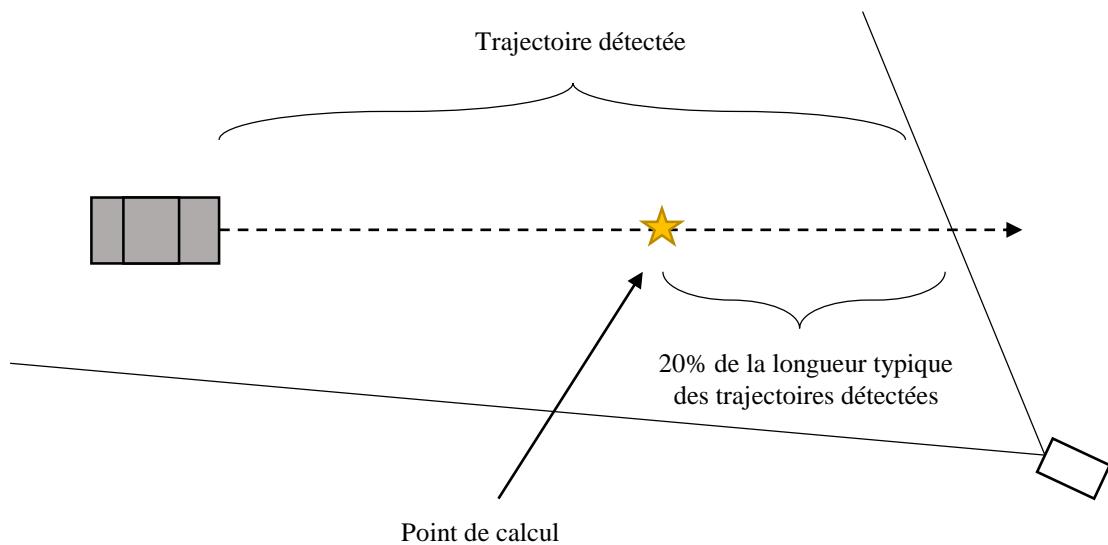


Figure 3-3 : Illustration du point de calcul des TIV de suivi

### 3.2.1.2 Les types de changements de voie

Pour l'étude, une distinction doit être faite entre les changements de voies requis pour respecter la destination voulue d'un conducteur, ci-après nommés des *changements de voie obligatoires*, et ceux effectués pour conserver une vitesse voulue par le conducteur, qui seront désignés comme *changements de voie opportunistes*.

La distinction est importante, puisque certains des modèles de simulation, comme VISSIM, donnent la priorité à l'atteinte de la destination des usagers, quitte à s'arrêter pour effectuer des changements de voie obligatoires, alors que les changements opportunistes dépendent uniquement des comportements des conducteurs et de la circulation (PTV, 2013). D'autres modèles, par exemple dans le logiciel Aimsun (TSS, 2008), permettent aux véhicules de ne pas suivre leur itinéraire initial s'ils ne sont pas capables de compléter les changements de voie requis. À ce moment, le modèle calcule un nouvel itinéraire. Pour ces modèles, la distinction entre les types de changements de voie serait à la fois moins pertinente, puisque tous les changements de voie sont traités de la même façon, et moins facile à faire, puisque plusieurs des critères présentés ici-bas ne sont fiables que si les changements de voie obligatoires ont toujours lieu.

Un comportement relaxant les contraintes de sécurité prises en compte dans le modèle, et donc une diminution de la taille des créneaux recherchés par le conducteur, ce qu'on appellerait un comportement plus agressif sur la route, se traduira par un plus grand nombre de changements de voies opportunistes effectués pour les mêmes conditions de circulation et les mêmes combinaisons de paires origine-destination. Le but est ici de tenter de reproduire ce nombre de changements de voies effectués par les conducteurs, puisqu'ils affectent les véhicules suiveurs qui doivent alors réagir à ce nouveau venu dans leur voie de circulation.

Pour calculer si un changement de voie est opportuniste ou obligatoire, la notion de « corridor » doit ici être définie. Cette notion permet de séparer les différents segments du réseau modélisé en groupes faisant partie d'un même élément du réseau et pouvant être étudié en un bloc cohérent.

Pour VISSIM, ceci permet de bien modéliser le réseau en regroupant les connecteurs et les liens faisant partie d'une même route lors des calculs. Par exemple, lorsqu'une voie de décélération apparaît avant une sortie sur une section d'autoroute, il est requis de séparer ce segment afin de permettre les changements de voie entre toutes les voies de l'autoroute et la voie de décélération pour que les comportements des véhicules puissent représenter la réalité. Ainsi, pour une autoroute

à deux voies, un premier lien de deux voies serait suivi d'un lien de trois voies, auquel seraient connecté à un lien de deux voies et un lien d'une voie (la sortie comme telle), tel qu'illustré à Figure 3-4. Ces différents liens doivent ensuite être reliés par des connecteurs, qui sont encadrés en jaune dans l'image. Dans cet exemple, un premier corridor comprendrait les deux liens de 2 voies et le lien de 3 voies ainsi que les connecteurs les reliant comme étant l'autoroute. Un deuxième corridor, correspondant à la sortie, comprendrait le lien d'une voie et le connecteur le reliant au lien de 3 voies.



Figure 3-4 : Sortie d'autoroute incluant une voie de décélération dans VISSIM

La logique utilisée pour déterminer si un changement de voie donnée est opportuniste ou obligatoire s'effectue en amorçant l'analyse de la trajectoire d'un véhicule avec le dernier changement de voie détecté et en classifiant chaque changement de voie précédent séquentiellement. Le concept de corridor nous permettant de séparer des zones du réseau, et donc de délimiter l'autoroute et sa sortie comme deux routes différentes, le premier critère utilisé est que l'ensemble de la trajectoire s'effectue ou non dans un même corridor. Si tel est le cas, comme dans la Figure 3-6 a, tous les changements de voies sont classifiés comme opportunistes sans avoir besoin de critères supplémentaires.

Dans le cas contraire, en commençant par le dernier changement de voie, nous qualifions les changements précédent selon deux critères principaux. Premièrement, la direction du changement de voie permet facilement de discriminer tout changement de voie éloignant le véhicule de la sortie (par exemple un véhicule effectuant un changement de voie vers la gauche alors que la sortie se trouve à droite) comme étant opportuniste. Dans le cas des changements de voie effectués dans la même direction que la sortie, l'état du changement de voie subséquent est appliqué à ce changement de voie. Ainsi, si le changement de voie à qualifier est suivi d'un changement de voie opportuniste, alors il est qualifié d'opportuniste, alors que s'il est suivi d'un changement de voie obligatoire, il est classifié comme obligatoire. Il est à noter que la classification du changement de voie #1 dans la Figure 3-6 b est en fait indéterminé, puisqu'il nous est impossible de connaître le processus mental du conducteur. Celui-ci aurait très bien pu vouloir se diriger vers sa sortie, puis, ayant l'espace, décidé d'effectuer un dépassement avant de ré-effectuer son changement de voie obligatoire. La classification est donc arbitraire, mais il était nécessaire de faire un choix. La méthodologie est résumée dans la Figure 3-5.

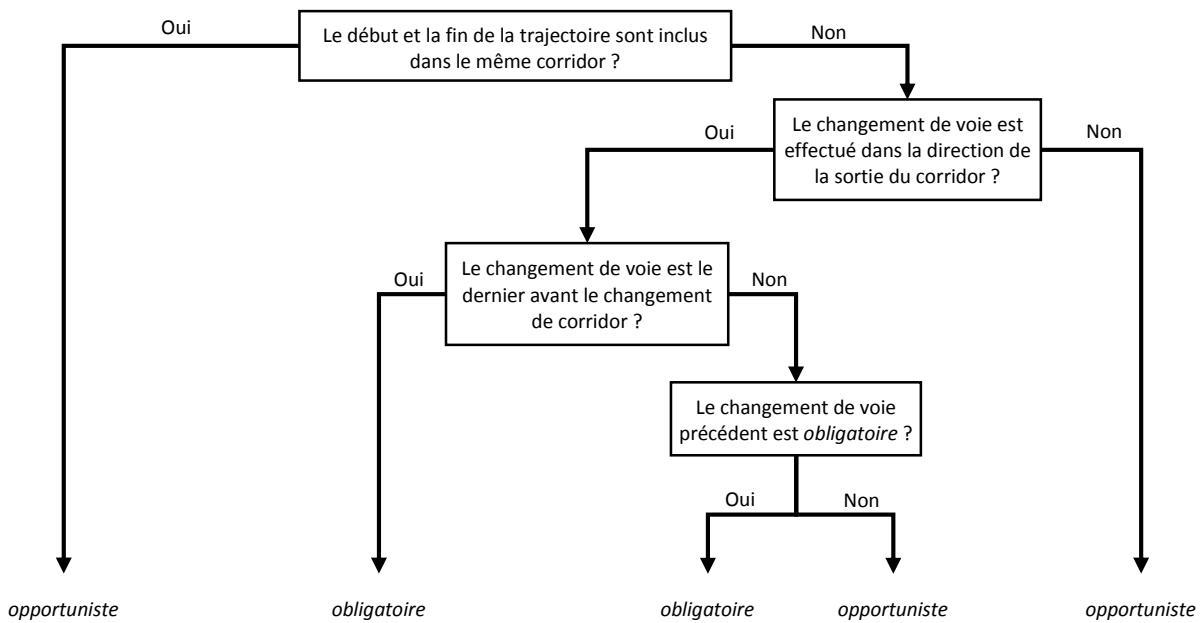


Figure 3-5: Arbre de classification des changements de voies, utilisé pour classifier chacun des changements de voie d'une trajectoire dans l'ordre chronologique inverse en commençant par le dernier.

Cette méthodologie est suffisante pour classer tous les changements de voies effectués dans les réseaux qui nous intéressent au court de ce projet de calibration. La notion de corridor s'applique aux autoroutes ou aux routes nationales sans croisement, chaque direction de la route constituant un corridor distinct. En milieu urbain, ou pour les segments de routes nationales avec carrefours à niveau, la définition présentée n'est pas directement applicable et il serait plus judicieux de définir une autre méthode de division du réseau, par exemple les approches d'un carrefour, et d'apporter quelques modifications à la méthodologie de classement des types de changements de voie présentée ici.

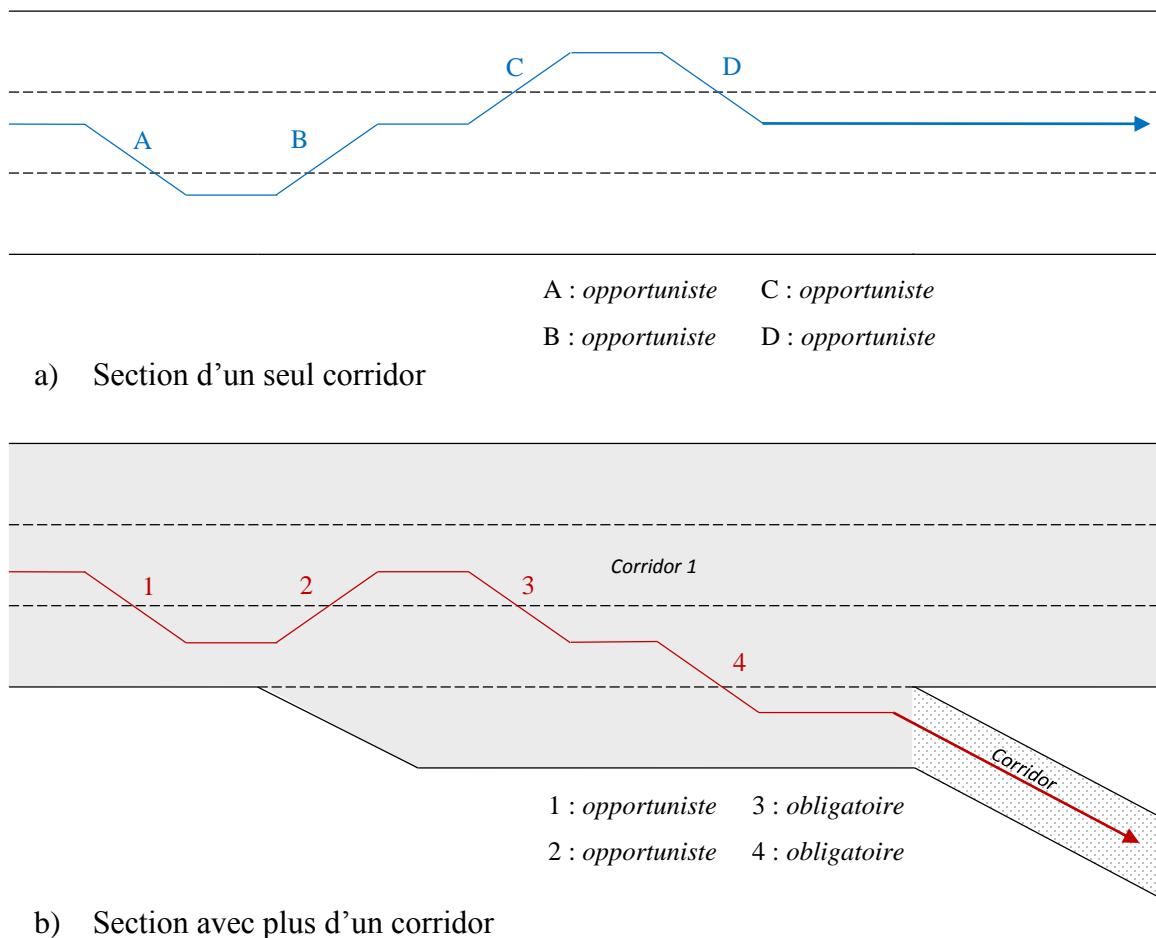


Figure 3-6 : Exemple de classification des changements de voies

### 3.2.2 Choix de l'algorithme d'optimisation

Le choix de l'algorithme d'optimisation doit se faire en considérant plusieurs critères, tels que discuté à la section 2.3, soit la capacité de l'algorithme à fournir une réponse qui soit un optimum local, le nombre de simulations nécessaire pour que l'algorithme chemine jusqu'à un optimum, la facilité d'utilisation et le comportement de l'algorithme sur le problème à calibrer. Pour le problème de calibration de modèles de simulation, la première considération est que l'algorithme choisi doit pouvoir écarter les combinaisons de paramètres créant des simulations aux conditions indésirables. Opérationnellement, cela signifie que l'algorithme doit permettre l'utilisation de contraintes de calibration. Ensuite, comme il sera discuté plus loin, chaque évaluation prend du temps à compléter. Par conséquent, l'algorithme doit minimiser autant que possible le nombre d'évaluations nécessaires. Finalement, les algorithmes avec preuve de convergence sont préférés aux heuristiques, qui ne nous garantissent pas que la solution soit un optimal local.

Au regard de ces critères, l'algorithme du *Mesh Adaptive Direct Search* (MADS) a été sélectionné (Audet & Dennis, 2006). MADS est un algorithme d'optimisation sans dérivés spécialisé pour effectuer l'optimisation avec contraintes de boîtes noires. Il procède en discrétilisant l'espace de recherche formé par les  $n$  paramètres de la boîte noire en une grille, appelée treillis. Il évalue ensuite avec la boîte noire (en fait l'adaptateur de la boîte noire) les candidats pris sur ce treillis. Chaque itération de MADS se décompose en une phase de recherche, le « *search* » et en une phase de sondage, le « *poll* ».

Durant la phase de recherche, l'algorithme fait une recherche globale dans l'espace discrétilisé, c'est-à-dire que les points générés sont sur le treillis mais pas nécessairement voisins de la solution courante. Cette recherche est effectuée dans l'espoir qu'un des candidats ainsi testé démontre la présence d'une région contenant de meilleure solution que celle explorée par la phase de sondage. Plusieurs stratégies peuvent ici être utilisées, dont des stratégies d'échantillonnage de l'espace ou utilisant des modèles de prédiction statistique. Une autre option possible, qui a été utilisée dans le cadre de ce projet se nomme la « *Variable Neighborhood Search* » (VNS) ou recherche à voisinages variables dont le but est de s'échapper des optimums locaux. Pour y arriver, la solution courante est perturbée en étant légèrement modifiée sur une ou plusieurs de ses dimensions. De ce point, une recherche locale, similaire à la phase de sondage de MADS est effectuée. L'amplitude de la perturbation est agrandie jusqu'à l'épuisement du budget d'évaluation ou la découverte d'un

nouveau meilleur point. Si la fin du VNS aboutit à un nouveau meilleur point, MADS reprend la recherche à partir de ce point et l'amplitude du VNS est remise à sa valeur initiale jusqu'à ce qu'un nouvel optimum soit atteint et que le VNS soit réactivé (Audet, Béchard, & Le Digabel, 2008).

La phase de sondage est plus rigidement définie puisque c'est elle qui permet à l'algorithme de posséder une preuve de convergence. À chaque itération de l'algorithme, la phase de sondage génère des candidats sur le treillis situés en deçà d'une certaine distance, appelée paramètre de sondage. Pour déterminer ces points, l'algorithme utilise des directions dans l'espace dont l'ensemble normalisé à la propriété est d'être dense sur la sphère unitaire.

Une fois les deux phases terminées, l'algorithme évalue si un succès a été obtenu en comparant la valeur de la fonction objectif des points testés à celle de la solution courante ainsi que leur respect des contraintes en utilisant la technique de la barrière progressive (Audet & Dennis, 2009). En cas de succès, soit que l'un des candidats testés améliorait la meilleure solution connue, le point de départ de la prochaine itération est déplacé sur ce point et le pas du treillis est agrandie. En cas d'échec, le point courant reste le même et le pas du treillis est réduit. Près d'un optimum local, des échecs répétés amèneront éventuellement la taille du treillis à zéro. Cette propriété est à la base de la preuve de convergence de l'algorithme, qui stipule que sous certaines hypothèses, la convergence globale (signifiant que le point de départ dans l'espace n'affecte pas la capacité de trouver un optimum) sur un optimum local est assurée (Audet & Dennis, 2006).

En pratique, l'algorithme MADS est implanté dans le logiciel NOMAD (Le Digabel, 2011) qui est disponible sous licence libre LGPL (« *open source* ») en ligne au [www.gerad.ca/nomad](http://www.gerad.ca/nomad). C'est ce logiciel qui est inséré dans la case « algorithme » de la méthode présentée à la Figure 3-1.

### **3.2.3 Méthode de collecte des données**

L'équipement utilisé pour la collecte est une caméra GoPro Hero 2 munie d'un dispositif Wi-Fi et d'une batterie externe. La caméra est protégée par un boîtier destiné aux caméras de sécurité qui est ensuite montée sur un mât télescopique fabriqué sur mesure, tel que décrit dans (Jackson, Miranda-Moreno, St-Aubin, & Saunier, 2012) et montré à la Figure 3-7. Le tout peut être rapidement déployé sur les sites où l'on retrouve un poteau d'utilité publique, de préférence un lampadaire où un poteau de signalisation lumineuse.

La caméra et la batterie externe sont fixées à l'intérieur du boîtier par des bandes velcro. Il faut donc faire attention lors de la mise en place afin de conserver l'alignement de la caméra dans le boîtier et éviter qu'elle ne filme que l'intérieur du boîtier. Le dispositif Wi-Fi permet de contrôler la caméra à distance à l'aide de l'application GoPro, ce qui permet de voir ce qui est couvert par la caméra même lorsque le mât est pleinement déployé et ainsi de s'assurer que la caméra est correctement orientée.

- a) Installation du mât sur un fût d'éclairage    b) Gros plan sur le dispositif de fixation de l'A20*



Figure 3-7 : Installation du système de collecte vidéo sur un fût d'éclairage du Ministère des Transports du Québec aux abords de l'Autoroute A20 dans l'Ouest de l'île de Montréal

Les images prises par la caméra sont enregistrées sur une carte SD de 64 Go, ce qui permet, avec les réglages décrits un peu plus loin, de récolter environ 10 heures de vidéos. Avec le présent dispositif, la carte mémoire manque d'espace bien avant que la batterie externe ne se soit complètement déchargée. Le mât est composé de plusieurs sections de tuyau de fibre de verre qui s'emboîtent les unes dans les autres. Lorsque pleinement déployé, il est possible de hisser la caméra jusqu'à une hauteur d'environ 8 m. La section du bas comporte une selle en métal qui s'appuie sur le poteau. On peut ensuite fixer le tout avec des sangles pour que la base soit solidement attachée. Le mât est fixé sur toute sa longueur sur le poteau en se servant de deux portions de corde fixées

sur la caméra et entouré autour du poteau. De cette façon, la caméra reste relativement stable même dans un vent de force moyenne. Des chaînes et un collier de serrage permettent de bloquer en place le dispositif contre d'éventuels vandales. Le dispositif, lorsque bien installé, a l'avantage d'être discret, ce qui évite qu'il soit remarqué par les usagers de la route. Un dispositif trop visible pourrait en effet nuire à la collecte de donnée en dérangeant les conducteurs et en modifiant leur comportement de conduite. Il se déploie par deux personnes en 15 à 30 min et se transporte facilement dans un VUS équipé de gyrophares et d'une flèche de signalisation selon les normes du Tome V – Signalisation routière du MTQ pour la sécurité de l'équipe d'installation.

Pour la collecte, la caméra a été configurée avec les options vidéo suivantes : résolution de 960 x 30 (soit 1280 x 960 pixels et 30 images par seconde) et un angle de vue « *Wide* ». Cette résolution présente un bon compromis entre la taille (en Go) de la vidéo et la qualité de l'image qui est cruciale pour que l'algorithme d'analyse vidéo détecte et suive bien les différents véhicules. Toutefois, une erreur de manipulation terrain a fait en sorte que la vidéo sur l'autoroute A15 a été enregistrée avec une résolution de 1080 x 30 (soit 1920 x 1080 pixels et 30 images par s). L'erreur n'est pas problématique en soit, tant que la bonne transformation est donnée pour l'analyse vidéo, puisque la résolution d'image est supérieure. Cependant, le fichier vidéo prend par conséquent plus d'espace et le nombre d'heures captées est donc réduit. La caméra utilisée crée automatiquement des séquences vidéo d'environ 20 à 21 min.

Le traitement des données vidéo se fait à l'aide du logiciel Traffic Intelligence, développé par le professeur Nicolas Saunier et des contributeurs externes. Ce logiciel permet de détecter et de suivre les usagers de la route en mouvement pour extraire leurs positions à chaque image. Dans un premier temps, le fonctionnement de Traffic intelligence sera explicité, puis les particularités liées au projet de calibration seront détaillés.

### **3.2.4 Traitement des données vidéos**

#### **3.2.4.1 Fonctionnement de Traffic Intelligence**

Le logiciel Traffic Intelligence fonctionne selon un principe de reconnaissance de trajectoires de points particuliers (*feature-based tracking*). Afin de bien fonctionner, le logiciel a besoin d'informations sur la vidéo, afin de projeter les positions de l'espace image dans un plan du monde réel. Cela permet d'effectuer les calculs de distance adéquatement avec les distances réelles. Pour

des caméras GoPro, l'objectif à grand-angle (*fish-eye*) crée une distorsion radiale qui doit d'abord être corrigée, afin que les coordonnées dans l'image puissent être projetées à l'aide d'une homographie (matrice 3x3) dans le plan du monde réel pour l'analyse.

À chaque image de la vidéo, le logiciel balaie l'image et tente de trouver la nouvelle position des pixels qu'il a reconnus dans les images précédentes. Si des pixels sont trouvés dans une certaine zone de l'image autour de leurs positions précédentes, ils sont conservés pour créer des trajectoires de ces points particuliers (Saunier & Sayed, 2006).

Les trajectoires de points particuliers sont ensuite groupées pour former des objets. Les critères permettant de relier ces trajectoires de points particuliers sont cruciaux. Puisque plusieurs erreurs peuvent survenir à cette étape, il faudra par la suite les éliminer dans le post-traitement. Le premier critère est que la trajectoire de points considérés doit être présente sur plus de trois images (critère pouvant être changé par l'utilisateur) et que le déplacement du point suivi doit respecter un certain minimum, fixé par l'utilisateur. Elle doit également être présente sur un nombre minimum d'images en même temps que d'autres trajectoires de points particuliers déjà attribués à l'objet. De plus, elle doit être comprise à l'intérieur d'une distance maximale des autres trajectoires de points déjà attribués à l'objet. Ensuite, pour chaque paire de trajectoires, le déplacement relatif des points est calculé et si celui est trop grand, l'algorithme sépare alors les trajectoires en deux objets (Saunier & Sayed, 2006). Des trajectoires de points particuliers peuvent être perdues ou s'ajouter à l'objet en tout point de la vidéo. Les causes peuvent être un objet masquant la vision de la caméra, par exemple un poteau, ou encore un véhicule effectuant un virage et où le côté initialement suivi deviendra graduellement masqué pour la caméra par un côté de ce même véhicule.

Le produit final de l'algorithme est une base de données SQLite comportant les positions des objets pour chacune des images de la vidéo analysée. Le temps est donc calculé en image, soit 1/30 de seconde pour la résolution choisie des GoPro Hero 2. Une méthode par défaut d'obtenir les positions successives des objets est de calculer le centre de masse des trajectoires de points particuliers à chaque instant.

Quelques problèmes avec l'algorithme sont connus et nécessitent de traiter les trajectoires produites lors de l'analyse de la vidéo. Premièrement, le centre de masse peut être instable ou non représentatif de la position réelle du véhicule (Saunier & Sayed, 2006). Ce problème, illustré à la Figure 3-8, est dû au fait que d'une image à l'autre au cours de la vidéo, le nombre de pixels suivis

ainsi que leur position peut être variable. Comme le centre de masse est situé au centre du polygone passant par tous les pixels détectés, sa position est affectée par des pertes et des gains de pixels. Dans certains cas, un groupement peut s'effectuer sur une zone plus précise, par exemple le haut de l'objet, tel que montré en  $t+1$  dans la Figure 3-8. Ce phénomène peut même mener à l'illusion qu'un véhicule recule momentanément si un groupement vers l'avant du véhicule est rapidement remplacé par un groupement vers l'arrière de celui-ci entre quelques images successives. Ce problème touche particulièrement les camions, puisque ceux-ci présentent une plus grande surface de détection et donc une plus grande possibilité que des pertes ou des gains de pixels déplacent le centre de masse ou encore qu'un nouveau groupement se fasse. Le problème existe également au début et à la fin des trajectoires, alors que le véhicule apparaît et disparaît, mais puisqu'elle affecte un très grand nombre d'objets, l'erreur est plus facile à détecter et à prendre en compte.

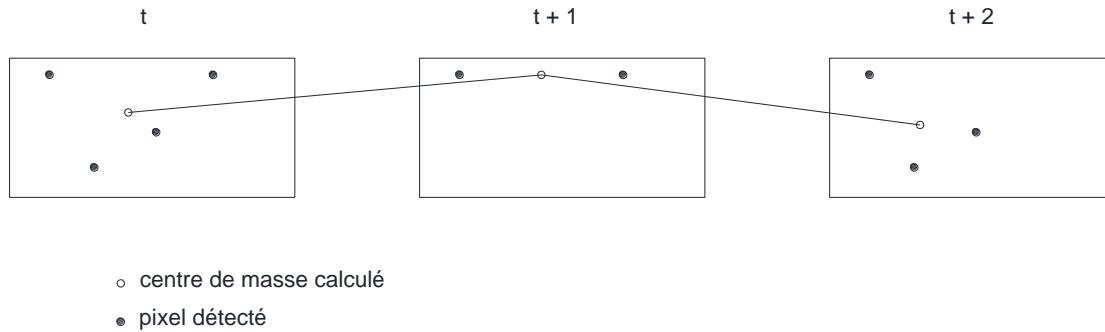


Figure 3-8 : Illustration du bruit enregistré sur la position du centre de masse d'un objet engendré par la perte et le gain de pixels d'une image à l'autre

Ce problème en engendre un deuxième, qui encore une fois est plus important dans le cas des camions, soit une augmentation de l'erreur de parallaxe lors de la projection du centre de masse au sol pour déterminer la position de l'objet, tel qu'illustré à la Figure 3-9. Cet effet est causé principalement par deux éléments : la position de la caméra et la hauteur de suivi sur l'objet. En effet, plus le centre de masse est détecté en hauteur, et plus le véhicule sera projeté loin. Pour une position donnée de la caméra, les différents types de véhicules subiront tous un certain déplacement latéral lors de la détection de leur position par rapport à leur position au sol réelle, qui serait obtenue si la caméra était placée directement au-dessus de l'objet. La conséquence est que toutes les trajectoires sont légèrement décalées : les conséquences de ce problème ne sont cependant pas si importantes si les véhicules subissent à peu près tous le même décalage, car la cohérence entre les

trajectoires et leurs positions relatives sont conservées. Or, dans le cas des camions qui seraient suivis très en hauteur, le déplacement causé par l'erreur de parallaxe peut faire en sorte que le véhicule semble être dans une voie de circulation autre que celle où il se trouve réellement. Ceci peut être visualisé à la Figure 3-10, où le camion circulant dans la première voie (la plus à droite) est projeté au même endroit que les véhicules circulants dans la troisième voie (la plus à gauche).

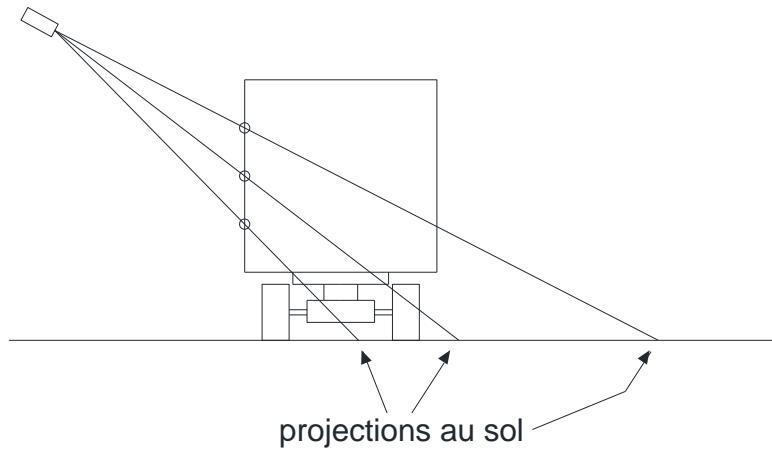


Figure 3-9 : Illustration de l'amplification de l'erreur de parallaxe engendrée par un centre de masse détecté sur le haut d'un camion



Figure 3-10 : Exemple d'un camion détecté très haut sur la boîte de sa remorque, engendrant une augmentation de l'erreur de parallaxe

De plus, certains véhicules, principalement les plus gros, peuvent être reconnus comme deux véhicules distincts. Certaines rares occurrences de véhicules entrant dans le champ de la caméra à

des instants très proches et ayant des trajectoires similaires peuvent également être jumelées, générant dans la base de données une trajectoire impossible dans la réalité. Certains objets peuvent également être masqués par un autre véhicule auquel ils auront été associés parce qu'un véhicule plus gros bloque le champ de vision de la caméra. Le placement de la caméra sur le site doit donc être fait de façon à minimiser les occlusions des objets fixes dans la scène et des véhicules entre eux, c'est-à-dire avec une vue plongeante et en angle par rapport à la route. Les erreurs sont également plus fréquentes lorsque les véhicules sont situés loin dans le champ de vision (Saunier & Sayed, 2006). Ces erreurs peuvent être réduites en réduisant la zone analysée de l'image. Finalement, la distorsion de la lentille doit être prise en compte pour réduire les erreurs de parallaxe.

Tout dépendant de l'analyse effectuée, certaines erreurs auront un plus ou moins grand effet sur les résultats. Par exemple, si on cherche à calculer des temps intervéhiculaires, les erreurs sur les positions du centre de masse sont importantes, alors que si on cherche à trouver un débit, ce sont plutôt les duplications ou les véhicules non suivis qui posent problème.

### **3.2.4.2 Détermination des vidéos conservées**

Afin d'assurer la qualité des données de référence, deux étapes de traitement doivent être effectuées. Dans un premier temps, un comptage manuel des véhicules pour les cinq et dix premières minutes de chacune des vidéos sur chacun des sites est fait. Ce comptage a été effectué par voie de circulation, ce qui est utile pour la procédure décrite dans la section 3.2.4.3, bien que pour cette étape du traitement, seule la somme des comptages sur les voies est utilisée. L'algorithme de détection et suivi de Traffic Intelligence est ensuite utilisé sur chacune des vidéos pour les cinq premières minutes, puis les dix premières minutes, et le nombre total de véhicules ainsi détectés est comparé au comptage manuel. La même analyse de Traffic Intelligence est utilisée pour compiler la vitesse moyenne des véhicules sur les vidéos, pour les cinq et dix premières minutes.

Le choix des vidéos est fait selon la logique suivante : toutes les vidéos dont l'écart entre les comptages manuels et automatiques était de 10 % ou moins pour les plages des cinq et dix premières minutes ont été conservées. Le graphique de la vitesse permet de qualifier les conditions de circulation (fluides ou congestionnées) sur une vidéo donnée puisque le débit ne le permet pas à lui seul. En combinant les informations, il devient possible de savoir quelles vidéos se trouvent à la fois dans des conditions congestionnées (faible vitesse) et avec peu d'erreurs de comptages, ou

non congestionnées (vitesses élevées) et avec peu d'erreurs de comptages, afin de sélectionner des conditions de circulation différentes lors des phases de calibration et validation.

La deuxième étape de traitement n'est effectuée que sur les vidéos conservées au terme de la première étape. Pour chacune des vidéos conservées, les trajectoires des véhicules ayant une vitesse négative ou une vitesse trop élevée dont le seuil doit être déterminé selon les conditions de collectes et le positionnement de la caméra (Anderson-Trocmé, Stipancic, Miranda-Moreno, & Saunier, 2015) sont sélectionnées pour inspection visuelle dans la vidéo, après laquelle les erreurs de suivi sont éliminées de l'ensemble de données.

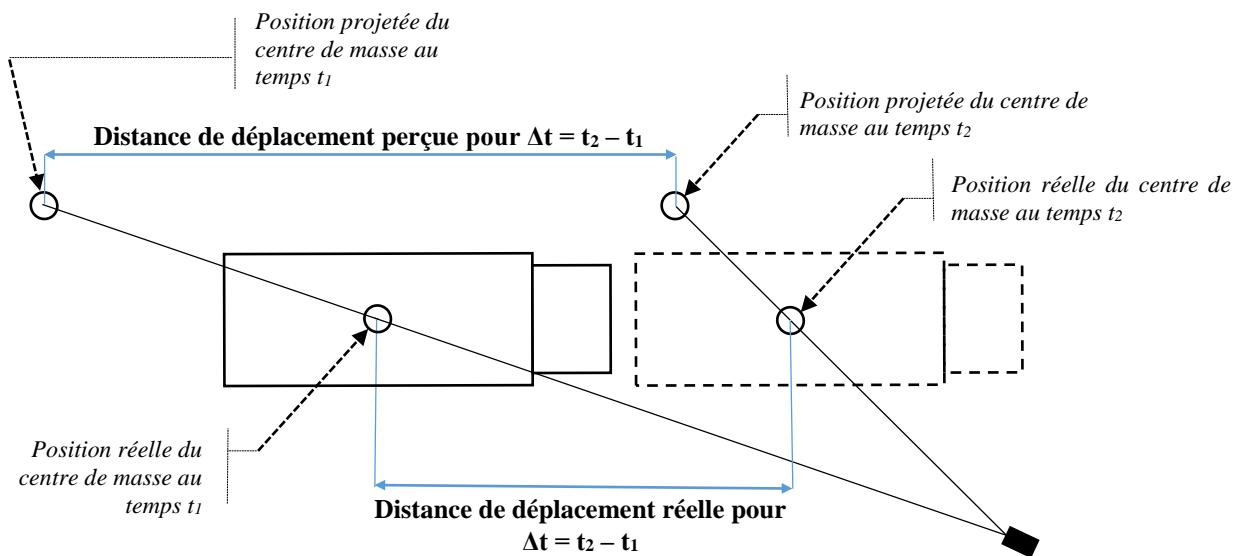


Figure 3-11 : Illustration de la cause des vitesses trop élevées calculées pour les objets ayant un grand effet de parallaxe dû à un suivi des points éloignés du sol

Les vitesses négatives indiquent des véhicules circulant en sens inverse, mais ayant été comptés dans la zone de calcul soit à la suite d'une erreur de projection de la caméra ou d'une mauvaise définition de la zone de traitement donnée à Traffic Intelligence. Dans tous les cas rencontrés, le véhicule devait être retiré de la banque de données de référence. Pour les vitesses trop élevées, un seuil, à partir duquel les vitesses observées sont jugées irréalistes, doit être choisi. Le critère menant à sa détermination est qu'il doit être tel que, à l'inspection visuelle, la totalité des véhicules détectés avec une vitesse supérieure ou égale au seuil soient des camions dont beaucoup de pixels situés sur le haut de la boîte soient enregistrés et très peu dans la partie près du sol. Comme expliqué plus

haut, cette situation amplifie l'erreur de parallaxe de la caméra et positionne donc le centre de masse du véhicule au mauvais endroit de la chaussée. Vue dans le plan horizontal, comme à la Figure 3-11, le centre de masse du véhicule semble se déplacer plus vite qu'il ne le fait réellement, puisque l'erreur de parallaxe sur la position du centre de masse affecte les deux dimensions du plan de détection parallèle au sol, soit les directions perpendiculaire et parallèles à la direction de la caméra. Cette erreur est facilement détectée par mesure de vitesses puisque la conséquence est que la vitesse calculée pour ces véhicules est anormalement élevée.

Finalement, des cas de très petits TIV indiquant bien souvent des doubles détections d'un même véhicule doivent être investigués.

### **3.2.4.3 Assignation des positions des véhicules à des voies de circulation**

La suite du traitement des données vidéo consiste à assigner les véhicules sur les voies de circulation. La procédure consiste à dessiner, grâce à une interface disponible dans CATS, une ou des polylignes représentant le centre des voies de circulation désirées. Il est alors possible de demander à Traffic Intelligence d'assigner toutes les positions des trajectoires à la polyligne la plus rapprochée. Un lissage est effectué afin d'éviter des sauts ponctuels de voies, donnant un résultat plus réaliste.

Une illustration de l'assignation est montrée à la Figure 3-12. Les centres de voie sont illustrés par les lignes pointillées. Chaque couleur correspond à la voie à laquelle les trajectoires véhiculaires sont assignées lors de l'entrée du véhicule dans le champ de vision de la caméra. Les débuts de trajectoire sont identifiés avec un point et on peut donc noter que les véhicules se dirigent du coin supérieur droit vers le coin inférieur gauche, soit du plus éloigné de la caméra vers le plus près de celle-ci, puisque la circulation dans les voies étudiées s'en rapprochait, tel que montré à la Figure 4-1.

La position exacte n'est pas facile à déterminer, puisque les positions de tous véhicules sont affectées par l'erreur de parallaxe. Ainsi, même le fait d'afficher l'image satellite afin d'obtenir des repères ne fournit qu'une approximation. Il a fallu procéder par itération afin de tester les emplacements des centres de voie pour obtenir une répartition des véhicules semblable à celle obtenue par comptage manuel. La procédure mériterait d'être développée afin de pouvoir prédire, lors du traçage d'une polyligne, comment se répartiraient les véhicules en plusieurs points et de comparer le tout avec un comptage en plusieurs points de la voie. Ce comptage pourrait également

être effectué par Traffic Intelligence en effectuant le suivi sur une mince bande, à l'instar des capteurs sous la chaussée et en optimisant les paramètres de l'algorithme pour minimiser les erreurs de suivi dans ces conditions. Malheureusement, ceci constituerait en soit un projet à part entière et n'a donc pas pu être implanté.

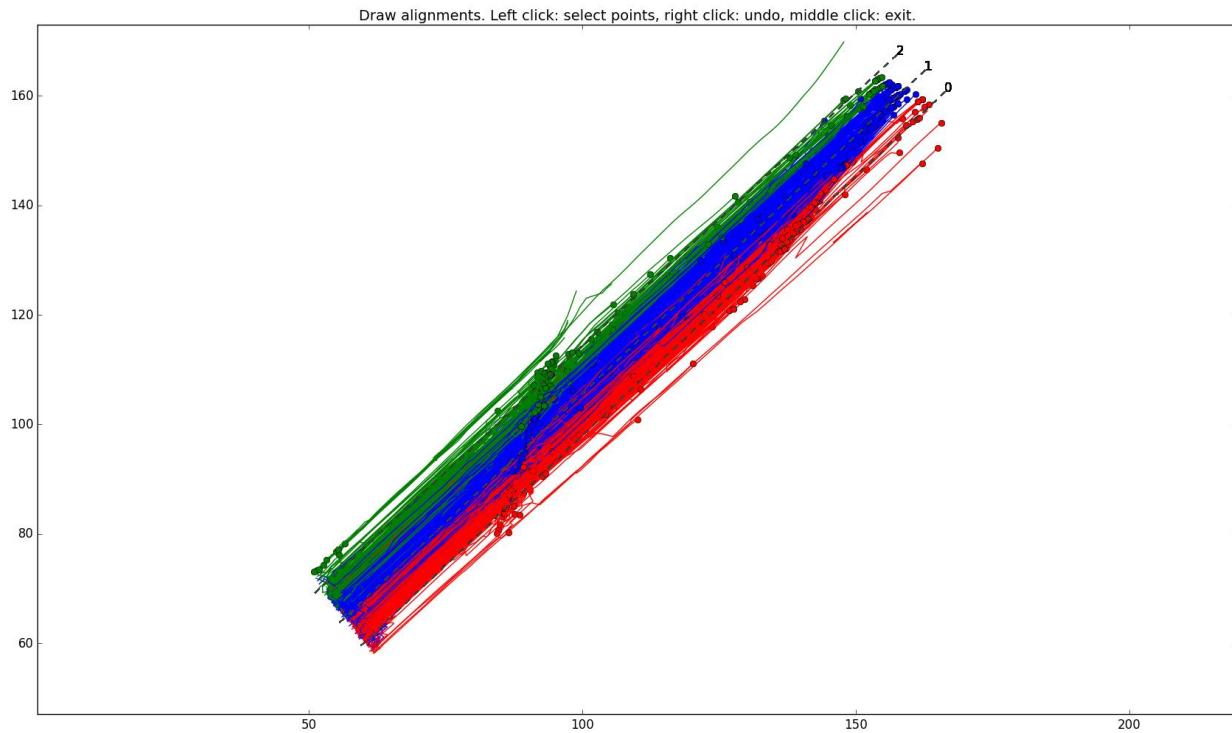


Figure 3-12 : Assignation aux voies des véhicules détectés durant les 5 premières minutes de la vidéo A13-10

Une fois cette dernière étape accomplie, les données de trajectoires des différentes vidéos ont été sauvegardées sous forme précalculée afin d'être rapidement chargées lors de la procédure de calibration.

### 3.2.5 Identification des paramètres et analyse de sensibilité

La première étape de la calibration consiste en la détermination des paramètres pertinents ainsi que des bornes à leur apposer. Un premier relevé des paramètres des deux modèles impliqués a permis de dresser la liste des paramètres pertinents pour la calibration. Ces valeurs sont présentées plus en détail avec les résultats de calibration au Chapitre 5. Certains paramètres ont des bornes fournies par VISSIM et leur violation entraîne une erreur critique du logiciel et l'arrêt de la simulation.

Évidemment, ces bornes sont considérées comme inviolables et sont restées fixées aux valeurs fournies par le manuel de VISSIM durant tous les essais de calibration.

Les autres bornes ont été déterminées selon un processus itératif. (Gamboa, 2014) a effectué la première étape de détermination, soit une revue des plages de valeurs les plus couramment utilisées pour ces différents paramètres dans la littérature et les différents guides de modélisation en vigueur dans le monde. De ce point de départ, des essais de calibration ont été effectuées. Une analyse de l'historique de calibration permet de visualiser si une des bornes semble arrêter la progression de l'algorithme, ce qui s'observe par une accumulation des points testés près d'une borne. Ceci peut être visualisé à la Figure 5-1, à la page 69. Lorsqu'une telle situation était observée, les bornes étaient éloignées pour la calibration suivante. Inversement, si une situation pathologique émergeait du fait d'une borne trop permissive, cette dernière était resserrée. Les valeurs montrées au Tableau 5-1 sont donc les valeurs finales utilisées lors des calibrations ayant mené aux résultats présentés au Chapitre 5.

À ce point-ci, il aurait été judicieux d'effectuer une analyse de sensibilité afin de réduire le nombre de paramètres impliqués dans la calibration. Une première tentative a d'ailleurs été conduite dans (Gamboa, 2014). Toutefois, cette étude se limitait à une analyse des paramètres un à la fois et les effets possibles de synergie entre les paramètres ne pouvaient alors pas être étudiés. Comme le plus gros ensemble de paramètres à calibrer ne comporte que 20 paramètres, l'algorithme de calibration peut traiter l'ensemble des paramètres. Il a donc été jugé préférable de tous les conserver plutôt que de risquer d'en éliminer sans connaître leur effet complet, quitte à ce que la calibration soit plus lente à converger du au plus grand espace à explorer. De plus, NOMAD est conçu pour adapter sa recherche à la sensibilité perçue des différentes variables et ce n'est donc pas problématique de hausser le nombre de paramètres calibrés simultanément jusqu'à un tel chiffre.

Cette dernière décision conclut la préparation de la calibration en fournissant la dernière portion de l'identification du problème de calibration, après l'identification des indicateurs et la collecte de données. Les sections suivantes couvrent les décisions en lien avec la calibration et la validation.

### 3.3 Phase de calibration

#### 3.3.1 Choix de la fonction d'ajustement

Les distributions statistiques des TIV simulés sont comparées à l'aide du test de Kolmogorov-Smirnov. Ce test non paramétrique peut être utilisé pour évaluer si un échantillon suit une loi donnée ou si deux échantillons sont issus de la même loi. Pour ce faire, le test évalue la proximité entre les fonctions de répartition de l'échantillon et de la distribution théorique, ou encore les fonctions de répartition de deux échantillons. La Figure 3-13 illustre cette distance par une flèche noire située à la distance maximale entre les courbes des fonctions de répartition de la loi normale calculées avec des paramètres différents.

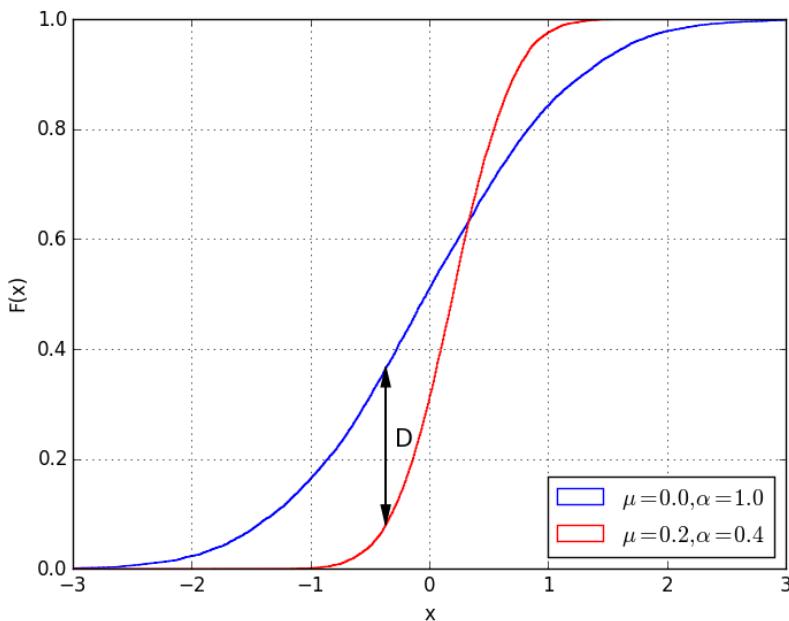


Figure 3-13 : Illustration de la statistique D du test de Kolmogorov-Smirnov

La version retenue du test de Kolmogorov-Smirnov est celle comparant deux échantillons, soit celui issu de la modélisation et celui issu de l'observation vidéo. La statistique du test, notée D, est la valeur absolue de la différence maximale entre les fonctions de répartitions (CDF) qui peut ensuite être utilisée pour calculer la probabilité de se tromper en rejetant l'hypothèse nulle que les deux échantillons sont issus de la même loi (risque de première espèce). Lorsque D = 0, on a une superposition parfaite des deux fonctions de répartition et le risque de première espèce du test est 1. Dans le cas d'une calibration où nous cherchons à diminuer l'écart entre les distributions

observées et simulées, il est plus simple de travailler directement avec la valeur D comme fonction d'ajustement.

### **3.3.2 La calibration sur plusieurs sites et vidéos**

Afin d'éviter le sur-ajustement de la calibration et de permettre que les paramètres calibrés soient utilisables sur le plus de sites et dans le plus grand nombre de conditions possibles, la calibration est effectuée simultanément en utilisant plusieurs vidéos et plusieurs sites lorsque cela est possible.

Les deux stratégies peuvent être combinées, jusqu'à un total de quatre réseaux différents en même temps. Ici, un réseau fait référence à un fichier VISSIM devant être simulé, et non nécessairement à des sites différents. Ainsi, si des conditions de circulation différentes doivent être simulées, elles doivent être incluses dans des fichiers VISSIM différents, qui comptent alors vers la limite de quatre. La raison de cette limitation est due aux limitations de la licence VISSIM utilisée, qui rend le temps de calcul prohibitif si cinq sites ou plus sont utilisés. Cette limite pourrait être relevée en faisant l'acquisition de licences supplémentaires. Le nombre de vidéos par site est quant à lui illimité.

Pour assurer la progression de la meilleure solution possible sur toutes les données utilisées, la fonction d'ajustement décrite à la section 3.3.1 est calculée pour chaque vidéo. La fonction objectif finale communiquée à NOMAD est le maximum entre les solutions calculées pour chacune des vidéos utilisées dans la calibration.

Cette stratégie s'assure que la calibration progresse en parallèle sur chacun des sites sans aller dans une direction optimisant les résultats d'un seul site au détriment d'un autre, puisque NOMAD ne reçoit comme information que la pire de toutes les solutions. Ainsi, nous pouvons avoir confiance que la solution trouvée sur chacun des sites est au moins aussi bonne que celle trouvée sur le pire site et permet donc d'étendre la plage des valeurs trouvées pour les paramètres calibrés sur laquelle nous avons confiance. La faiblesse de cette stratégie, comme nous le verrons dans la section des résultats, est si un des indicateurs utilisés dans la fonction maximum n'est pratiquement pas amélioré, puisque l'algorithme de calibration n'a aucune information sur la progression des autres indicateurs.

### 3.3.3 La fonction objectif

Afin de prendre en compte à la fois les TIV de suivi et les TIV acceptés pour prendre en compte l'ensemble des modèles (de poursuite et de changement de voie) s'appliquant à une section donnée d'autoroute, la fonction objectif  $f$  repose sur des calculs de D (Kolmogorov-Smirnov) sur les deux types de distributions de TIV. Afin de forcer une amélioration sur les deux types de données, la fonction retourne ensuite le maximum des deux valeurs observées.

$$f = \max(g_1, g_2)$$

avec :  $g_j = \max_i(d_{i,j}) \quad i \in \{1, 2, \dots, m\}, j \in \{1, 2\}$

et  $d_{i,j} = \sup_x |F_{i,j,1}(x) - F_{i,j,2}(x)|$

où  $i$  est le numéro de la vidéo,  $F_{i,j,1}(x)$  est la CDF des TIV de type  $j$  extraits des données vidéos,  $F_{i,j,2}(x)$  est la CDF des TIV de type  $j$  extraits des données simulées,  $g_1$  et  $g_2$  sont respectivement les sous-fonctions objectifs reliées aux TIV de suivi et aux TIV acceptés.

## 3.4 Phase de validation

La phase de validation des paramètres calibrés est conceptuellement très proche de la phase de calibration. Le schéma présenté à la Figure 3-1 est repris avec deux légères modifications. La première différence est l'absence de l'algorithme d'optimisation et la deuxième est l'ouverture de la boucle de rétroaction : une seule évaluation des paramètres issus de la calibration est effectuée.

Pour effectuer la validation, des nouveaux jeux de données sont utilisés, mais les indicateurs et la fonction objectif sont les mêmes et le nombre de répétitions effectuées pour tester le jeu de paramètres calibrés est le même que pour tester les candidats lors de la phase de calibration. La validation permet d'assurer que les paramètres obtenus à la suite du processus de calibration ne sont pas trop adaptés aux jeux de données utilisés pour la calibration, perdant du même coup leur capacité à bien décrire les conditions de circulation autre que celles observées dans ces vidéos de calibration.

Si les résultats de la validation sont concluants, c'est-à-dire qu'ils satisfont aux contraintes de calibration et donnent un résultat acceptable lors de l'évaluation de la fonction objectif, la

calibration est considérée un succès et les paramètres calibrés deviennent des paramètres validés. Dans le cas contraire, c'est tout le processus de calibration qui doit être revu, avec des modifications permettant d'améliorer le produit calibré, selon le critère ayant permis de rejeter la calibration. C'est cette grande boucle de rétroaction qui a permis de construire les contraintes, de raffiner les bornes imposées aux différents paramètres de VISSIM ainsi que de déterminer la combinaison de vidéos à utiliser pour assurer que les résultats soient valides dans le plus de conditions possibles.

## **CHAPITRE 4 CAS D'ÉTUDE**

Le présent chapitre traite des phases de calibration et de validation en lien avec l'utilisation de VISSIM pour la calibration du modèle de Wiedemann 99. Les données de calibration ont été collectées sur l'autoroute A13. Les différents choix nécessaires à l'application de la méthodologie générale présentée au Chapitre 3 pour le problème de calibration des modèles de conduite pour les autoroutes montréalaises y sont discutés.

### **4.1 Phase de préparation**

#### **4.1.1 Collecte et traitement des données vidéos**

##### **4.1.1.1 Détermination des sites de collecte**

Avant de commencer cette section, il faut mentionner que le projet initial visait la calibration des deux modèles de poursuite de VISSIM : Wiedemann 74, qui est recommandé dans les sections d'entrées et de sorties autoroutières de même que dans les zones urbaines, et Wiedemann 99, qui est recommandé dans les sections autoroutières sans entrecroisement. La collecte de donnée reflète donc le besoin de données pour la calibration de ces deux modèles. Toutefois, des discussions avec la Ville de Montréal en cours de projet ont mené à la décision de ne calibrer que le modèle de Wiedemann 99 dans le cadre de cette étude. C'est pourquoi le reste de ce qui est présenté dans ce mémoire ne mentionne pas les décisions à prendre en lien avec la calibration du modèle de Wiedemann 74.

La collecte de données a eu lieu entre le 22 mai et le 17 juin 2014, sur approximativement 10 heures consécutives en chaque site, de sorte que l'heure de pointe soit capturée. Les jours choisis étaient en semaine avant le déclenchement des vacances estivales et par temps dégagé. La prise de vue obtenue sur chacun de ces sites est montrée à la Figure 4-1, montrant le champ de vision à chacun des sites. Chaque site a fait l'objet de demandes d'entraves auprès des autorités responsables et un camion atténuateur d'impact accompagnait l'équipe terrain lorsque la collecte se faisait directement à partir de l'autoroute.

Pour mener à bien le projet de la Ville de Montréal, quatre types d'infrastructures ont été identifiées. Afin de calibrer le modèle sans convergence (Wiedemann 99), une section autoroutière

à deux voies et une section autoroutière à trois voies ont été sélectionnées afin de voir si les comportements des usagers changent entre ces deux types d'aménagements. En théorie, puisque ces sections ne sont pas, par définition, influencées par des entrées ou des sorties, on ne devrait pas observer une grande différence de comportement. Or, dans le réseau montréalais, les autoroutes à deux voies se retrouvent en périphérie de l'île et sont beaucoup moins achalandées que les autoroutes à trois voies qui composent l'épine dorsale du système routier supérieur montréalais. Dans ces conditions, il pourrait y avoir un changement de comportement et il était plus prudent de vérifier les conditions dans les deux types d'infrastructures.

a) A13 Nord



b) A15 Nord



c) A20 Est



d) A25 Sud



Figure 4-1 : Les différents lieux de collectes vues selon la caméra déployée



Figure 4-2 : Vue aérienne des différents sites de collecte, emplacement et champs de vision approximatif des caméras

Pour le modèle avec convergence (Wiedemann 74), deux types d'aménagements ont également été identifiés. Ici aussi une section à deux voies et une section à trois voies ont été sélectionnées. Pour la section à trois voies, il a été décidé de collecter de l'information sur une section de convergence, soit à une entrée d'autoroute. Pour la section à deux voies, une section d'entrecroisement, soit une section d'autoroute comprise entre une entrée et une sortie rapprochée a initialement été identifiée. Nous verrons toutefois que des contraintes de placement de la caméra ont fait en sorte que le champ de vision de la caméra était plutôt centré sur la fin de cette section et que nous capturons donc principalement une zone de divergence.

Tableau 4-1: Informations relatives à la collecte des données

<i>Lieu</i>	<i>Date</i>	<i>Type d'infrastructure</i>	<i>Vitesse maximale</i>	<i>Modèle*</i>	<i>Pointe</i>
A13 Nord, sortie 8	11 juin 2014	Section droite à trois voies,	100 km/h	W99	PM
A15 Nord, sortie 66	22 mai 2014	Section de converge à trois voies	70 km/h	W74	AM, PM
A20 Est, sortie 50	6 juin 2014	Section d'entrecroisement	100 km/h	W74	AM, PM
A25 Sud, rue Larry	17 juin 2014	Section droite à deux voies	100 km/h	W99	AM

\*W74 = Wiedemann 74; W99 = Wiedemann 99

Une fois les types d'aménagements désirés identifiés, les sites potentiels ont été explorés sur le réseau montréalais. Plusieurs sections ont d'emblée été exclues dû à la difficulté d'y installer l'équipement. C'est le cas notamment de toutes les sections surélevées de l'autoroute A40 et de l'autoroute A720. Les sites situés trop près d'un chantier de construction ont également été écartés afin d'éviter de capturer des comportements altérés par des conditions de circulations non standards. Vu le nombre de chantiers en cours à Montréal lors de la période de collecte, cette dernière contrainte a réduit de beaucoup le nombre de sites disponibles. Grâce à Google Street View, des sites potentiels ont pu être identifiés et une tournée terrain a permis de confirmer les emplacements finaux. Autant que possible, les sites nécessitant de se rendre directement sur les terrains du MTQ ont été écartés puisqu'il fallait alors procéder à des entraves temporaires nécessitant l'intervention d'un atténuateur d'impact fixé sur un véhicule (AIFV), ce qui fait considérablement augmenter les coûts de la collecte. De plus, les plages horaires acceptées par le ministère pour effectuer ces installations sont entre 23 h et 5 h, ce qui exclut de capturer la pointe de soir par manque d'espace de stockage sur la caméra.

Le Tableau 4-1 résume les différents lieux de collecte ainsi que les types d'aménagements étudiés et le modèle correspondant à calibrer et ceux-ci sont illustrés à la Figure 4-2.

#### 4.1.1.2 Détermination des vidéos conservées

Premièrement, puisque seul le modèle de Wiedemann 99 est calibré, seules les vidéos des sites qui avaient été ciblés à cette fin, soit ceux de l'A13 et de l'A25, peuvent être conservés. Malheureusement, le site de l'A25 a dû être écarté, parce que le débit est si faible qu'aucune interaction entre les véhicules n'est visible dans les données récoltées. Il ne reste donc que les vidéos de l'A13 qui peuvent être utiles.

La première étape des traitements décrits à la section 3.2.4.2 peut être visualisée à la Figure 4-3, qui montre la comparaison des temps calculés manuellement et automatiquement pour les cinq premières minutes des vidéos, alors que la Figure 4-4 montre ceux de la comparaison sur les 10 premières minutes.

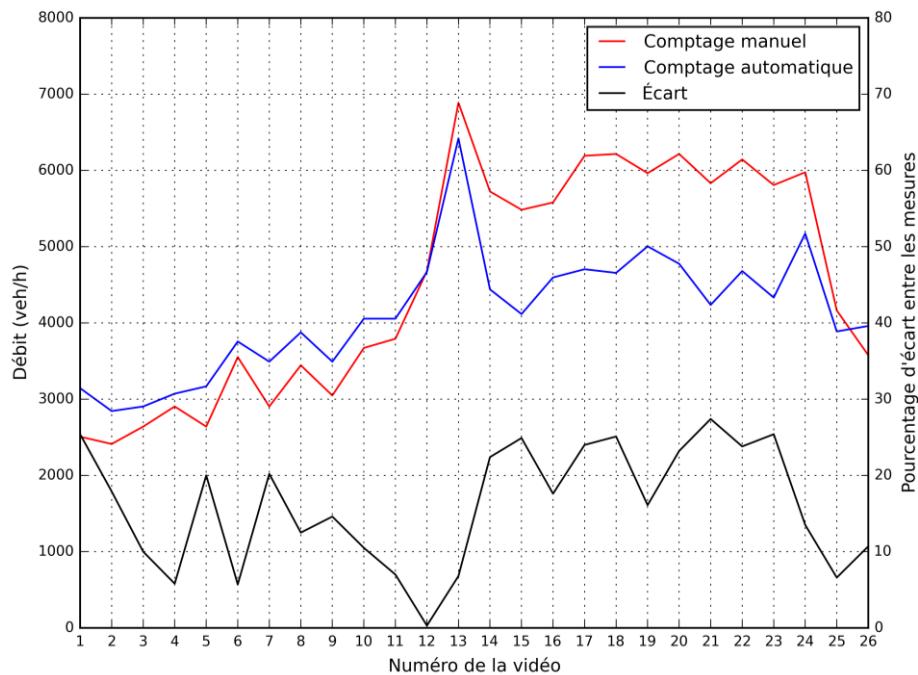


Figure 4-3 : Débit observé (et écart relatif) dans les cinq premières minutes des vidéos de l'A13 selon un comptage manuel et le comptage automatique de Traffic Intelligence

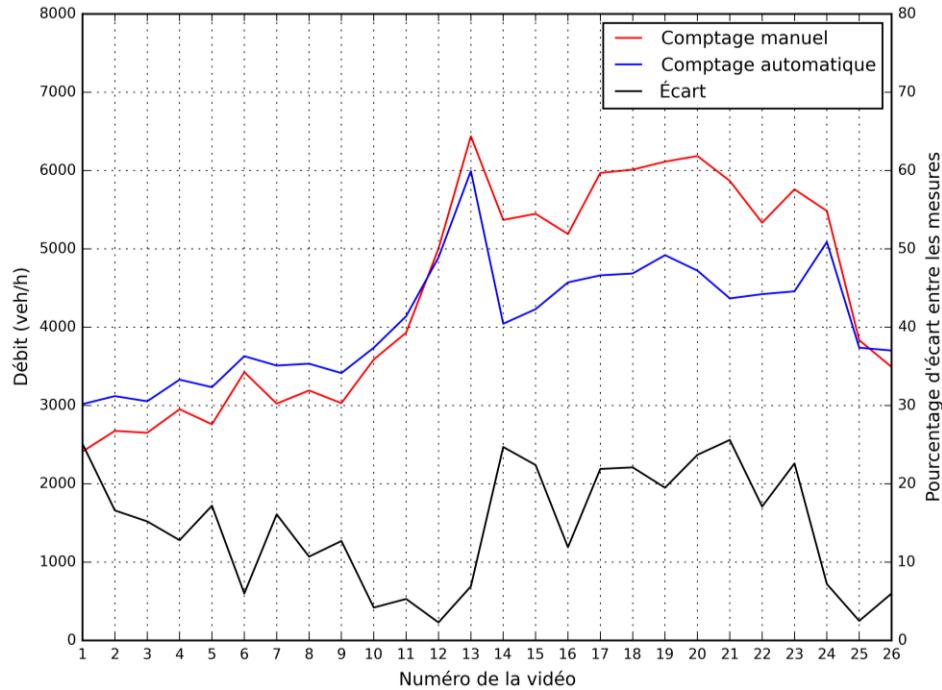


Figure 4-4 : Débit observé (et écart relatif) dans les dix premières minutes des vidéos de l'A13 selon un comptage manuel et le comptage automatique de Traffic Intelligence

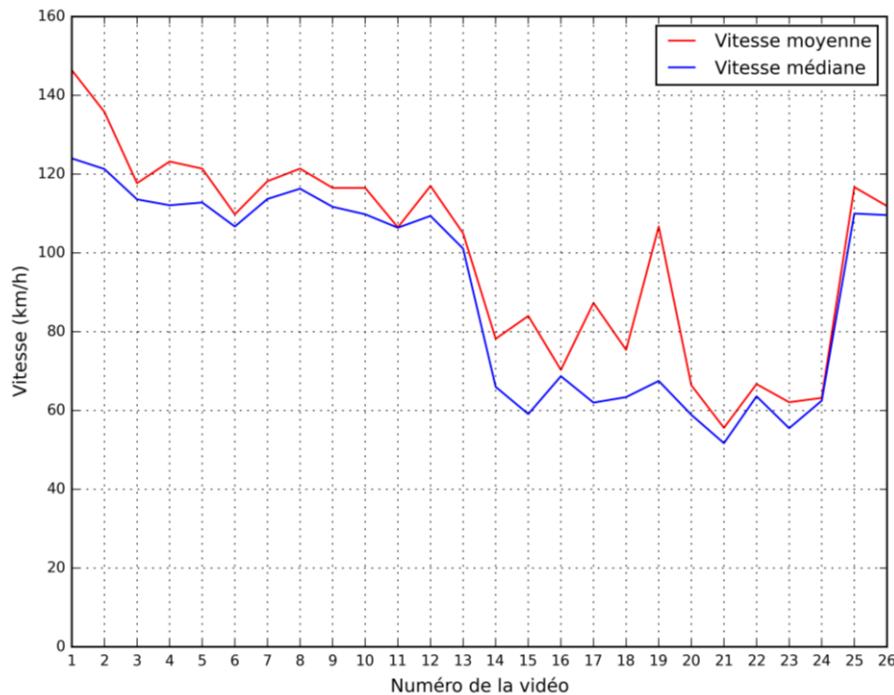


Figure 4-5 : Vitesses moyennes et médianes observées durant les cinq premières minutes des vidéos de l'A13, après réduction de 14 %

Une analyse similaire des résultats de Traffic Intelligence a compilé la vitesse moyenne des véhicules sur les vidéos, pour les 5 et 10 premières minutes. Pour cette figure, aucun écart relatif n'est fourni puisqu'aucune donnée comparative ne permettait d'apprécier l'erreur commise dans l'analyse automatique. Grâce aux travaux de (Anderson-Trocmé et al., 2015), il a été possible de déterminer les conditions de la pose de caméra afin d'appliquer un facteur de correction des vitesses. Par contre, le facteur de correction dépend de la vitesse observée. En l'absence d'une autre source de données de vitesse, il est difficile d'obtenir avec certitude le bon facteur de correction. La position de la caméra sur les différents sites étant plutôt défavorable pour l'extraction des vitesses, il a été jugé préférable d'appliquer un facteur constant de réduction de 14 % à l'ensemble des données et les résultats sont illustrés par la Figure 4-5. Cette correction, suggérée par (Anderson-Trocmé et al., 2015) pour le positionnement de notre caméra lors de la collecte, a été calibrée après la comparaison des données automatiques avec des données réelles. La vitesse n'étant par la suite utilisée que pour la distribution des vitesses désirées des véhicules générés dans la simulation VISSIM, nous avons jugé que cette erreur n'aurait qu'un impact minime sur la qualité du processus de calibration.

Les données incluses dans les Figure 4-3 et Figure 4-5 permettent de tracer le diagramme fondamental de la vitesse moyenne en fonction du débit pour cette portion d'autoroute, tel qu'illustré à la Figure 4-6.

Pour la deuxième étape de traitement, le seuil des vitesses trop élevé a été fixé à 150 km/h. Ce seuil a été choisi afin de conserver les véhicules roulant entre 120 km/h et 130 km/h, ce qui, bien qu'illégal, n'est pas inusité ou hors du commun. En majorant cette vitesse 14 %, afin d'être sur la même échelle de vitesse surestimée que dans les vidéos, nous obtenons environ 150 km/h.

Les véhicules détectés avec des vitesses trop élevées ont été retirés de l'ensemble de données, puisque l'erreur sur la position n'a pas qu'un effet sur les vitesses, mais également sur l'interaction que ces objets sembleront avoir avec les véhicules les entourant lors de l'analyse. Évidemment, le fait de les retirer augmente les temps intervéhiculaires mesurés pour les véhicules qui les suivent, mais le faible nombre de camions ainsi problématiques par rapport à l'ensemble total réduit grandement la portée de cette erreur. Le Tableau 4-2 présente l'importance de ces erreurs par rapport au nombre total de véhicules détectés dans les vidéos de l'A13 et montre que leur nombre est relativement limité. Le portrait est semblable pour les autres sites.

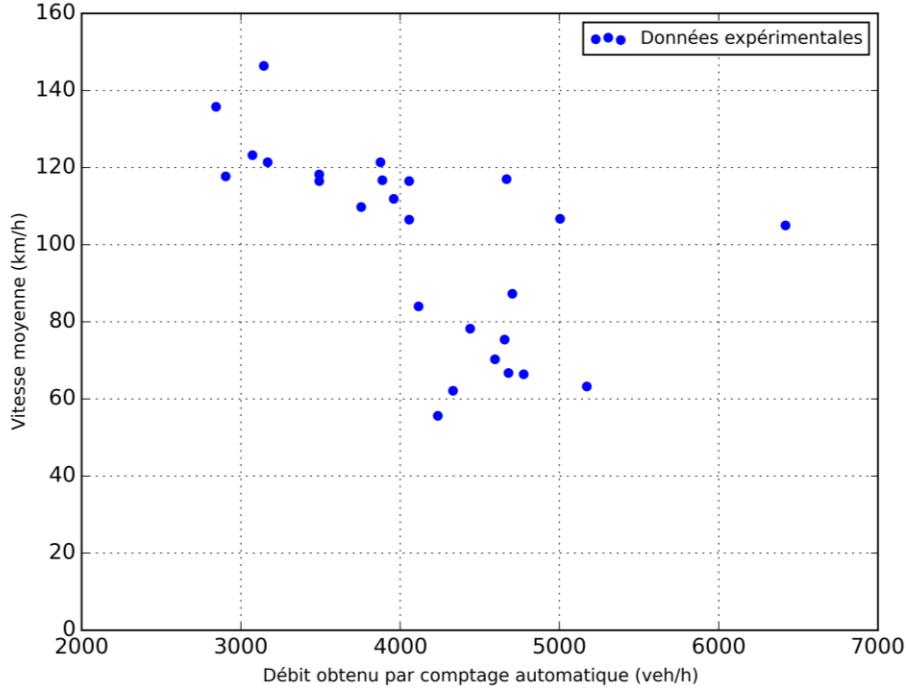


Figure 4-6 : Diagramme fondamental tracé pour le débit et la vitesse moyenne réduite de 14% des 5 premières minutes des vidéos, par intervalle de 20 minutes, pour la section étudiée de l'A13 selon les données expérimentales

Tableau 4-2: Importance des véhicules dont le suivi est problématique sur l'ensemble des véhicules détectés dans les vidéos de l'A13

Vidéo	Nombre total de véhicules	Nombre de véhicules problématiques	Pourcentage d'erreur
<i>Vidéos utilisées en calibration</i>			
A13-06	313	7	2,2
A13-11	338	14	4,1
A13-12	389	16	4,1
A13-13	535	16	3,0
<i>Vidéos utilisées en validation</i>			
A13-10	310	8	2,6
A13-25	324	13	4,0

Une autre erreur commune qu'il a fallu retirer est le suivi multiple d'un véhicule (sur-segmentation). Plusieurs facteurs peuvent expliquer l'apparition de cette erreur, qui consiste à avoir

différentes parties d'un même véhicule considérées comme des véhicules différents par l'algorithme. Cette erreur est particulièrement importante à minimiser dans notre cas puisqu'elle résulte en de très petits TIV, ce qui vient fausser leur distribution. Par observation des vidéos, il a été trouvé que n'importe quel TIV de 0,5 seconde ou moins correspondait à une erreur de suivi multiple et ces TIV ont été retirés de l'échantillon.

#### **4.1.2 Construction du modèle VISSIM**

Les modèles VISSIM utilisés sont basés sur le fichier de base de la Ville de Montréal, qui contient les données relatives aux accélérations des véhicules observés sur le réseau routier québécois. Le réseau de l'A13 a été construit à partir de ce fichier en suivant la géométrie du site couvert par la collecte de données.

Le réseau a été étendu en amont de près du double de la portion visible dans le champ de la caméra et a été étendu légèrement en aval. Cet allongement permet aux véhicules d'interagir avant et après le point de collecte d'information et d'ainsi assurer que les véhicules auront eu le temps d'interagir selon les jeux de paramètres testés pour avoir un effet sur les distributions étudiées au point de calcul. Les sections amont et aval sont sur des liens différents, reliés par des connecteurs, au lien à étude. Ceci permet de n'effectuer les calculs pertinents de temps intervéhiculaires que sur les liens étudiés pour économiser le temps de calcul. Comme expliqué dans la section 3.2.1.2, les liens pertinents sont ensuite groupés en corridors pour permettre la classification des types de changements de voies. Pour le cas d'étude, aucune sortie n'était incluse et par conséquent, un seul corridor a été créé. Seuls les changements de voie opportunistes sont donc comptabilisés.

Le réseau résultant étant très petit en longueur, le temps de simulation et de traitement des données reste modeste, ce qui permet de tester rapidement plusieurs jeux de paramètres lors de la calibration. Cette faible grandeur du réseau est expliquée par le champ de vision limité de la caméra : à peine une centaine de mètres où il est réaliste de collecter les données de circulation à l'aide de la caméra. L'allongement du réseau par rapport à ce champ de vision, soit la portion du réseau située en amont du viaduc du boulevard Henri-Bourassa sur l'image satellite de la Figure 4-7, ne sert qu'à stabiliser les comportements véhiculaires après leur génération sur le réseau.

Les données de circulation, soit la distribution des vitesses observées, le débit et le pourcentage de véhicules empruntant les entrées et sorties d'autoroute, lorsqu'applicables, ont été comptabilisées

lors du traitement des vidéos à l'aide de Traffic Intelligence, comme expliqué dans la section 3.2.4.2. Le pourcentage de véhicules lourds a été comptabilisé manuellement. Les débits, variables selon les vidéos utilisées, sont présentés avec les résultats, au Tableau 5-2.



Figure 4-7 : Modèle VISSIM de l’Autoroute 13

La Figure 4-8 montre la courbe de distribution cumulée des vitesses telle qu’entrée dans VISSIM. La distribution est celle des vitesses individuelles réduite du facteur de correction de 14%. Elle a été calculée grâce à l’ensemble des vidéos conservées et est donc constante pour tous les jeux de données testés et pour chacune des vidéos analysées. Ce choix a été fait afin d’assurer que les véhicules aient la possibilité d’avoir une vitesse désirée indépendante des conditions de circulation, en posant l’hypothèse que la congestion dans le modèle ramènerait cette vitesse à celle observée dans les différentes vidéos selon les frictions induites par le modèle de poursuite, une fois le processus de calibration terminé.

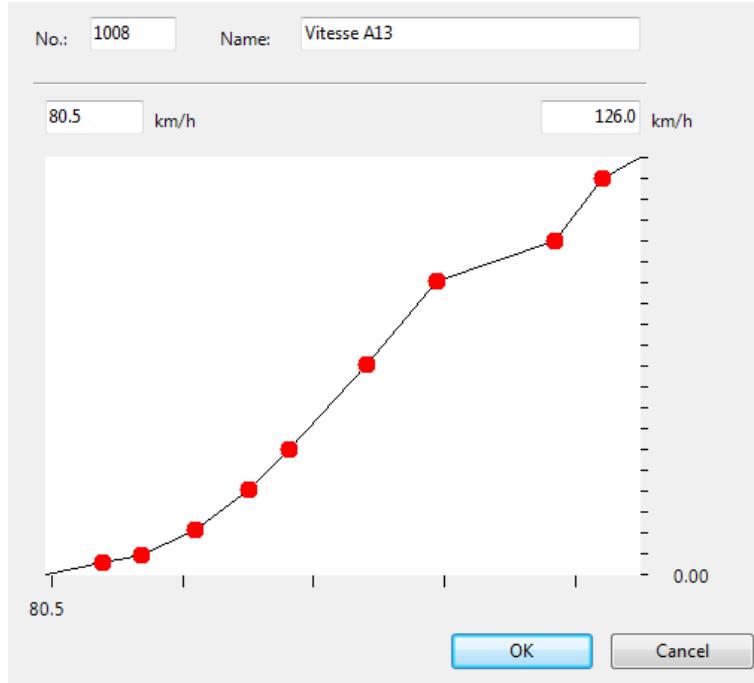


Figure 4-8 : Distribution cumulée des vitesses désirées telle qu'entrée dans le modèle VISSIM de la A13 (les vitesses sont représentées sur l'axe des abscisses et la probabilité cumulée sur l'axe des ordonnées)

Due à sa très grande simplicité, le réseau VISSIM utilisé dans cette étude n'a pas fait l'objet d'une recherche d'erreurs de codification extensive. Tout de même, des inspections visuelles ont été conduites dans les premières simulations pour assurer que le débit d'entrée sur le réseau passait sans que ne soient visibles des problématiques dues à la géométrie du réseau. Évidemment, cette étape pourrait être beaucoup plus longue pour un réseau complexe.

## 4.2 Phase de calibration

### 4.2.1 Formulation des contraintes

En optimisation, les contraintes permettent d'exprimer des régions de l'espace de solutions qui présentent des caractéristiques indésirables. Dans le cas de VISSIM, au fur et à mesure que nous progressions dans le travail de calibration, il nous a été possible d'observer que certaines combinaisons de paramètres aboutissent en des comportements véhiculaires inacceptables qui, poussés à l'extrême, engendrent le blocage complet du réseau et ce qui ressemble à des carambolages. Évidemment, ces situations peuvent exister sur le réseau réel dans des conditions

défavorables bien précises, mais elles ne sont pas souhaitables dans une modélisation visant à étudier et reproduire les conditions normales de circulation sur une infrastructure donnée.

La difficulté consiste à créer des indicateurs permettant de détecter ces situations et d'en fournir l'information à l'algorithme de calibration. Pour y arriver, deux stratégies, donnant quatre contraintes, ont été utilisées.

La première stratégie consiste à utiliser le fichier d'erreur généré par VISSIM lors de la simulation. Trois types d'erreurs ont été détectés comme indicatrices d'une situation anormale : l'erreur de génération des véhicules, l'erreur de décélération et l'erreur d'accélération. Les erreurs présentent les textes suivants, qui peuvent être reconnus directement dans le fichier d'erreur.

- *Vehicle input [I] could not be finished completely (remain: [V] vehicles).*
- *Simulation time [T] : The expected trajectory of vehicle [N] on link [L] at coordinate [S] cannot be determined because the expected deceleration is positive.*
- *Simulation time [T] : The expected trajectory of vehicle [N] on link [L] at coordinate [S] cannot be determined because the expected acceleration is zero.*

Les lettres T, I, L, S et V entre crochets sont des variables dont la valeur dépend de la simulation et de la cause de l'erreur : I, N et L étant des clefs d'identification de certains objets, V étant un comptage des véhicules n'ayant pu être générés dans le temps imparti pour la simulation et T et S étant les coordonnées spatio-temporelle de l'erreur.

Une lecture de chaque fichier d'erreur est effectuée pour quantifier l'importance de ces erreurs. La première est obtenue directement du texte d'erreur, la valeur V donnant le nombre de véhicules n'ayant pu être générés. Pour les deux autres, chaque apparition de l'erreur est comptée. Les fichiers d'erreurs de toutes les réplications en ayant généré sont comptabilisés et les résultats additionnés pour obtenir le nombre total d'erreurs de chaque type pour cette combinaison de paramètres.

Cette stratégie n'arrivant pas à détecter toutes les situations problématiques, une deuxième stratégie utilisant l'information des trajectoires a été ajoutée. En effet, plusieurs des situations problématiques observées comportent des dépassements sur une même voie de circulation, comme on peut le voir le graphique espace-temps présenté à la Figure 4-10. À l'extrême, un carambolage a lieu, comme à la Figure 4-9, et les véhicules finissent complètement arrêtés, mais se dépassent

tout de même à l'intérieur d'une même voie sur une certaine distance avant le point d'arrêt. Évidemment, pour qu'une telle situation ait lieu dans la réalité, il faudrait littéralement qu'un véhicule pulvérise le précédent pour passer au travers de lui. Cette situation ne devrait normalement pas être possible et l'on s'attendrait à ce qu'une vérification soit effectuée par le logiciel afin d'empêcher ces cas. Il s'agit donc ici d'une faiblesse du logiciel avec laquelle il est indispensable de travailler et il est important de repérer et d'éliminer ces situations puisqu'elles n'ont aucun sens physique par rapport aux conditions réelles de circulation.



Figure 4-9 : Exemple d'un carambolage causé par un mauvais jeu de valeurs pour les paramètres de modèle de poursuite Wiedemann 99

La Figure 4-10 montre le diagramme espace-temps sur la voie 1 du réseau simulant l'A13 à Montréal. On remarque clairement que des véhicules changent d'ordre ou se passent au travers l'un de l'autre (l'un passant de meneur à suiveur et l'autre de suiveur à meneur) sans effectuer de changement de voie, ce qui est marqué par le croisement de deux lignes. Évidemment, cette situation est impossible sur une voie réelle ayant des dimensions normales : le véhicule devrait normalement quitter la voie, effectuer son dépassement puis y revenir, ce qui se traduirait graphiquement par une ligne s'interrompant et une deuxième apparaissant plus loin à une coordonnée (temps, distance) supérieure.

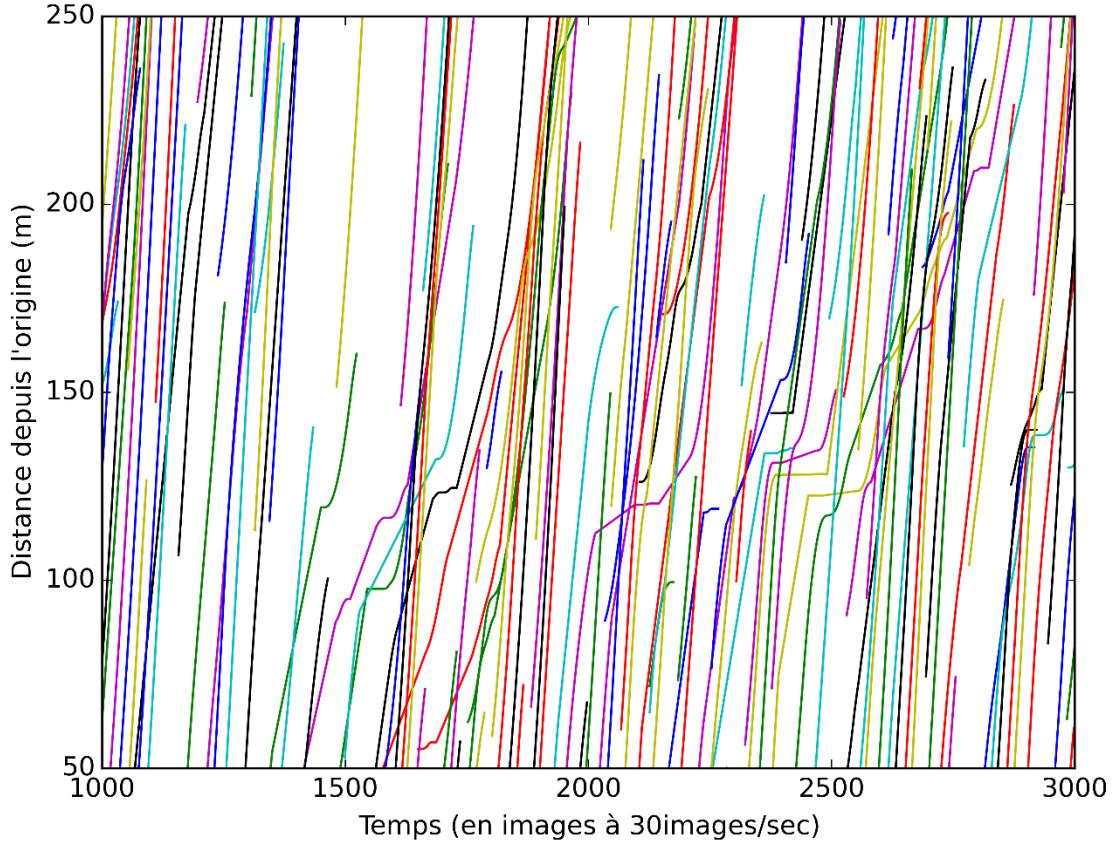


Figure 4-10 : Diagramme espace-temps de la première voie de circulation du réseau de l'A13 montrant des véhicules passant au travers du véhicule les précédant

Les trois premières contraintes ont été formulées de façon à être quantifiables. Pour chaque type d'erreur  $i$ , le nombre d'occurrences  $nb_{ij}$  de l'erreur est comptabilisé pour chaque réplication  $j$  du jeu de paramètres testé. Afin de « punir » les jeux de paramètres engendrant au moins une réplication donnant des résultats problématiques, le maximum d'erreur sur les  $n$  répétitions est retourné. Un seuil d'acceptabilité a été défini, au-delà duquel la simulation est jugée problématique. Puisqu'une situation acceptable est représentée par un résultat inférieur ou égal à 0, la formulation de ces contraintes est la suivante :

$$C_i = \max_{j=1 \dots n}(nb_{ij}) - seuil_i \leq 0; i = 1 \dots 3$$

Ainsi, si le nombre d'observations est inférieur ou égal au seuil, la valeur finale obtenue est négative ou égale à 0 et l'algorithme retient le point comme faisant partie du domaine réalisable. Inversement, si le seuil est dépassé, le résultat est positif et l'algorithme reçoit l'information qu'il

se trouve hors du domaine réalisable. L'avantage de ces trois contraintes est qu'elles sont quantifiables et peuvent par conséquent être traitées à l'aide de différentes méthodes de NOMAD de traitement des contraintes, bien que dans le cadre de cette étude seule la méthode de la barrière extrême ait été testée.

Des analyses empiriques des différentes simulations où les erreurs de modélisation étaient détectées ont mené à la détermination des seuils utilisés. Les valeurs retenues sont les valeurs minimales observées où les simulations étaient acceptables malgré la présence des erreurs. Il aura fallu plusieurs dizaines de points testés pour arriver à formuler ces seuils puisqu'il est rare que ces différentes erreurs arrivent seules. Les seuils acceptables finaux sont présentés dans le Tableau 4-3.

Tableau 4-3: Seuils associés aux contraintes utilisant le fichier d'erreur de VISSIM

<i>Erreur</i>	<i>Seuil</i>
“vehicle input could not be finished”	35
“deceleration is positive”	0
“acceleration is zero”	0

Malheureusement, la contrainte basée sur les positions des véhicules n'a pas pu être traitée de la même façon à cause de contraintes de calcul. En effet, lorsque les fichiers de trajectoires générés par VISSIM étaient trop volumineux (au-delà de 6 millions de lignes), la fonction de détection des situations problématiques causait une erreur de saturation de la mémoire vive qui causait alors une interruption du code. Cette interruption est par la suite interprétée par l'algorithme d'optimisation comme le fait que ce point est irréalisable, alors qu'il pouvait être un très bon point et même respecter les contraintes, quoique cette situation était peu probable puisque la taille du fichier est corrélée avec le fait que la simulation ait présenté une situation problématique. Évidemment, renvoyer la mauvaise information à l'algorithme de calibration est non souhaitable, car cette situation pourrait le priver de cheminer sur une trajectoire de calibration prometteuse.

Pour contourner ce problème, le fichier est coupé en plusieurs fichiers dont la grosseur est inférieure à 6 millions de lignes et chacun de ces fichiers temporaires est ensuite analysé. Le nombre de dépassements problématiques de chacun des fichiers est ensuite additionné. Or, le fait de couper le fichier prive parfois la fonction de détection de l'information nécessaire pour déterminer si un véhicule qui était suiveur auparavant est maintenant devenu meneur en passant au

travers du véhicule le précédent. Le problème est d'autant plus fréquent que le nombre de véhicules dans le réseau est important : les informations des positions successives sont alors plus espacées et donc plus susceptibles d'être séparées par la coupure du fichier. La sommation n'est donc pas toujours égale au nombre réel de dépassements problématiques ayant eu lieu et nous ne pouvons donc pas affirmer que la contrainte est quantifiable. Par conséquent, cette dernière contrainte ne peut être traitée que par une méthode de barrière extrême et aucun seuil d'acceptabilité n'a été déterminé : si au moins un dépassement problématique est relevé, l'algorithme reçoit l'information que le point est situé hors du domaine réalisable et ne s'aventurera jamais dans cette direction à cause de la barrière extrême. Il pourrait être intéressant ultérieurement de trouver une méthode de calcul permettant de traiter des fichiers plus gros, qui sont somme toute fréquents.

#### **4.2.2 Nombre de répétitions**

Puisque la fonction objectif sélectionnée se base sur des distributions statistiques et non sur leur moyenne, la méthode présentée à la section 2.1.2 utilisant la loi de Student n'est pas utilisable pour déterminer le nombre de répétitions appropriées pour assurer une confiance statistique dans la précision les résultats simulés. Pour pallier à ce problème, la stratégie présentée dans (Hollander & Liu, 2008a) est mise en œuvre afin de trouver le nombre de répétitions nécessaire pour que le résultat de la fonction objectif présentée à la section 3.3.1, qui utilise deux fois le test de Kolmogorov-Smirnov sur deux distributions de TIV distinctes, se stabilise. Les distributions de l'ensemble des répétitions sont ainsi mises ensembles pour former une seule grande distribution qui est comparée aux distributions observées dans les vidéos utilisées pour la calibration. Le test est effectué en utilisant les valeurs par défaut des paramètres impliqués dans la calibration (point de départ de l'optimisation).

La Figure 4-11 présente les résultats de ce test pour deux vidéos. Ces vidéos ont été choisies pour représenter les deux catégories de conditions routières présentes dans les vidéos, tel qu'on peut le remarquer au Tableau 5-2. Sur les courbes des deux catégories, il est possible de constater que malgré la volatilité des résultats observés pour chacune des répétitions individuellement, les distributions concaténant les données de ces répétitions se stabilisent assez rapidement, principalement dans le cas de la vidéo A13-13 qui est en circulation congestionnée. Pour cette vidéo, la valeur résultante est très stable dès 5 répétitions et varie très peu par la suite. Par contre, dans le cas de la vidéo A13-06, dont le débit de circulation est presque deux fois moins important,

la stabilisation se fait autour de la barre des 10 répétitions, possiblement vers 12 ou 13. Puisqu'il est important que cette stabilisation ait lieu pour toutes les vidéos, le nombre choisi doit donc être supérieur à 10 et dû à l'architecture de multicœur utilisé, il est plus efficace de choisir un multiple de 10. Ainsi, le nombre de répétitions effectuées pour chacun des jeux de paramètres testés dans le cadre de la calibration est de 20.

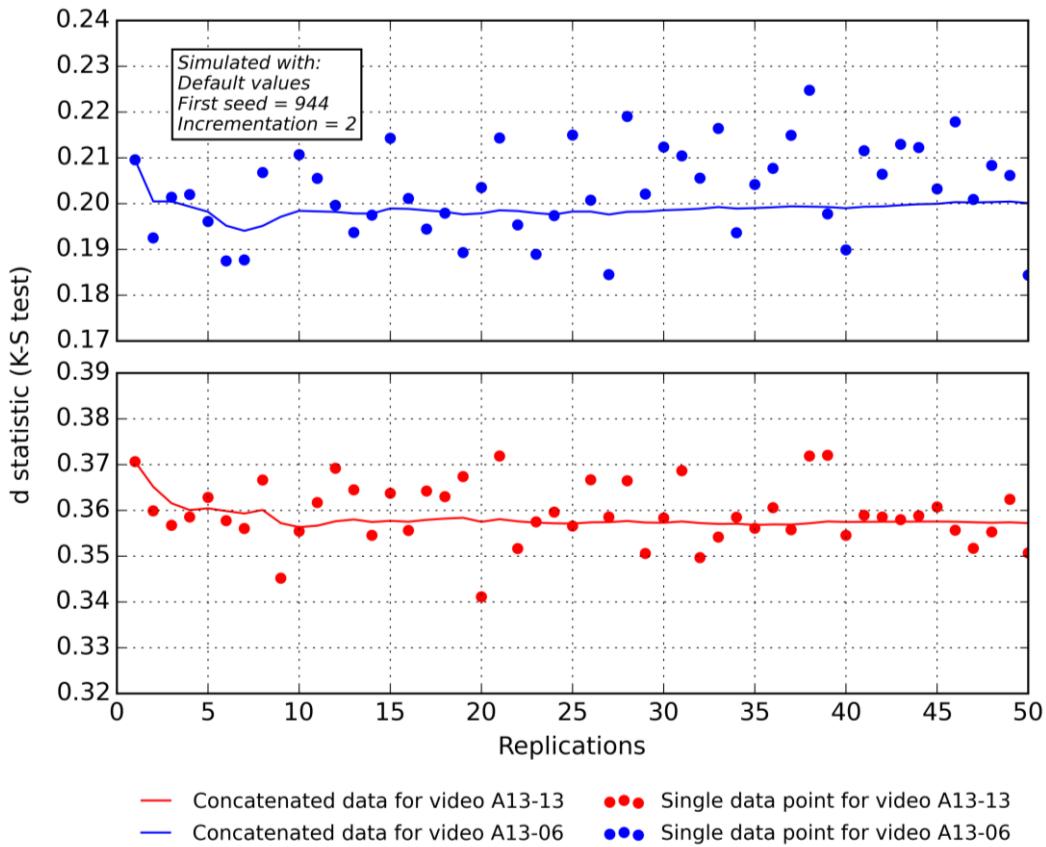


Figure 4-11 : Détermination du nombre de répétitions à effectuer selon une méthode de stabilisation de la valeur de la fonction objectif

### 4.2.3 Point de départ

Le point de départ utilisé pour la calibration est le vecteur constitué des valeurs par défaut proposées par le logiciel pour les différents paramètres inclus. Ce point est pratique puisqu'il constitue le point le plus souvent utilisé par les praticiens lors de leurs modélisations et il est donc intéressant de comparer les résultats de la fonction objectif entre ce point et le meilleur point trouvé au terme du processus de calibration. Le fait de l'inclure comme point de départ nous assure ainsi

d'explorer les environs du point le plus communément utilisé pour voir si certains des paramètres seraient déjà positionnés à une valeur plutôt optimale.

#### **4.2.4 Budget d'évaluation**

Avec tous les calculs requis pour l'évaluation des différentes parties de la fonction objectif, il faut entre 2 et 4 minutes pour compléter l'évaluation des 20 répétitions, et ce, pour chaque fichier Vissim impliqué. Puisque pour la calibration, quatre fichiers VISSIM, correspondants aux quatre vidéos utilisées pour la calibration, sont utilisés pour évaluer chacun des jeux de paramètres identifiés par NOMAD comme candidat potentiel, il devient vite très couteux en temps d'effectuer la calibration. Par conséquent, il est judicieux de limiter à un nombre raisonnable le budget alloué à NOMAD, tout en conservant un nombre permettant à l'algorithme de converger vers un optimum. Des essais successifs ont démontré que les points intéressants étaient trouvés autour de la marque des 900 points testés. Pour éviter de rater ce point, un budget de 1000 évaluations a été alloué au processus de calibration. En utilisant les estimations ci-haut, le processus peut prendre entre 5 et 11 jours à se compléter. Toutefois, les simulations problématiques et que les contraintes visent à éliminer peuvent être beaucoup plus longue à évaluer et certains essais de calibration ont pris jusqu'à un mois.

## CHAPITRE 5 RÉSULTATS

### 5.1 Calibration des paramètres

Le Tableau 5-1 présente les paramètres inclus dans le processus de calibration ainsi que les bornes inférieures et supérieures imposées à NOMAD dans la recherche du meilleur point, de même que le point initial, correspondant aux valeurs par défaut des paramètres, et le meilleur point trouvé. La très grande majorité des paramètres ont vu leur valeur varier substantiellement entre le point initial et le point calibré. À l’opposé, les paramètres w99cc0 et ObsrvdVeh n’ont vu aucune variation de leur meilleure valeur. L’illustration de l’historique à la Figure 5-1 montre d’ailleurs que les bornes n’ont pas bloqué outre mesure la recherche et que la faible variation de certains paramètres n’est pas due à une absence d’exploration.

Tableau 5-1 : Paramètres inclus dans l’étude : valeurs par défaut (point de départ), bornes supérieures et inférieures et valeur calibrée

<i>Paramètres</i>	<i>Valeur par défaut</i>	<i>Borne inférieure</i>	<i>Borne supérieure</i>	<i>Valeur calibrée</i>
<i>Modèle de poursuite</i>	w99cc0	1,5	0,0*	10,0
	w99cc1	0,9	0,0*	20,0
	w99cc2	4,0	0,0*	10,0
	w99cc3	-8,0	-30,0	30,0
	w99cc4	-0,35	-30,0	30,0
	w99cc5	0,35	-30,0	30,0
	w99cc6	11,44	-30,0	30,0
	w99cc7	0,25	-30,0	30,0
	w99cc8	3,5	0,1	120,0
	w99cc9	1,5	0,1	120,0
<i>Modèle de changement de voie</i>	ObsrvdVeh	2	0*	10
	MaxDecelOwn	-4,0	-10,0*	-0,02*
	DecelRedDistOwn	100,0	0,0*	200,0
	AccDecelOwn	-1,0	-10,0*	-1,0*
	MaxDecelTrail	-3,0	-10,0*	-0,02*
	DecelRedDistTrail	100,0	0,0*	200,0
	AccDecelTrail	-1,0	-10,0*	-1,0*
	DiffusTm	60,0	0,0	200,0
	MinHdwy	0,5	0,0*	10
	SafDistFactLnChg	0,6	0,0*	1,0*
<i>Valeur de l’objectif</i>	0,3601	---	---	0,3289
<i>Point réalisable ?</i>	Oui	---	---	Oui

Les bornes indiquées avec un astérisque (\*) sont des bornes imposées par le logiciel, leur non-respect entraînant une erreur fatale d’exécution

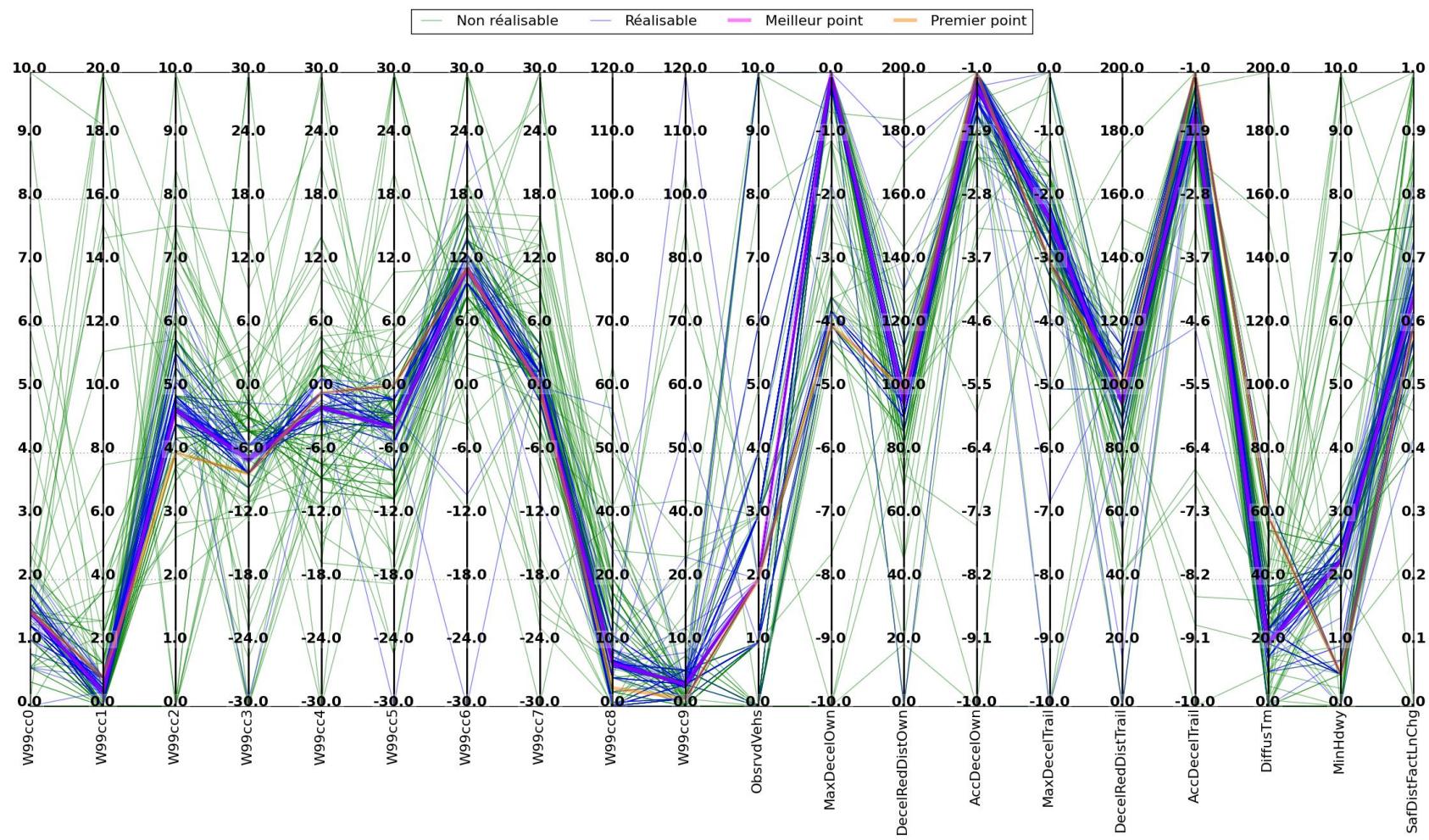


Figure 5-1 : Visualisation de l'espace des paramètres exploré par NOMAD

Un point sur le respect des contraintes qu'il n'est pas possible de représenter sur une figure est la propension des différents réseaux à violer les contraintes et à produire un point non réalisable dans la Figure 5-1. En effet, ce ne sont pas systématiquement les quatre simulations (soit une par vidéo) qui violent les contraintes alors que la seule différence entre ces simulations est le débit de circulation. La proportion exacte de points déclarés irréalisables par chacune des simulations n'a pas été comptabilisée, mais celui correspondant à la vidéo A13-13 est plus prompt à générer des situations problématiques. La raison exacte n'a pas été investiguée, mais une hypothèse est que le plus fort débit augmente le nombre d'interactions entre les véhicules et augmente la probabilité que plus d'un véhicule à la fois effectuent une manœuvre anormale (généralement un changement de voie directement devant un autre véhicule), menant à des erreurs du modèle et donc à la génération des fichiers d'erreur de VISSIM.

Tableau 5-2 : Débit de circulation et résultats de l'évaluation de la fonction objectif pour les paramètres par défaut et les paramètres calibrés pour les vidéos utilisées lors de la calibration et de la validation.

Vidéo	Débit (veh/h/ln)	Statistique D du test de Kolmogorov-Smirnov			
		Valeurs par défaut	Valeurs calibrées	TIV de suivi	TIV accepté
<b>Calibration</b>					
A13-06	1252	<b>0,1993</b>	0,1833	0,1265	<b>0,1743</b>
A13-11	1352	0,2994	<b>0,3136</b>	0,1916	<b>0,2906</b>
A13-12	1556	0,2896	<b>0,3601</b>	0,1639	<b>0,3213</b>
A13-13	2140	<b>0,3581</b>	0,3551	0,0820	<b>0,3289</b>
<i>Fonction objectif</i>	--		<b>0,3601</b>		<b>0,3289</b>
<b>Validation</b>					
A13-10	1352	0,1820	<b>0,3765</b>	0,1161	<b>0,3738</b>
A13-25	1296	<b>0,2197</b>	0,2068	0,1275	<b>0,2173</b>

Pour chacune des vidéos, le plus élevés des résultats au test de Kolmogorov-Smirnov est présenté en gras. La valeur de la fonction objectif est égale à la plus grande des valeurs en gras des quatre vidéos.

L'application de la méthode décrite précédemment donne des résultats mitigés, tel qu'il est possible de le constater dans le Tableau 5-2, présentant les résultats de l'évaluation de la fonction objectif au point initial et au point calibré. Il ressort clairement de ce tableau que la calibration des TIV de suivi a bien fonctionné, passant d'une valeur de 0,3581 dans le pire des cas avant la calibration à une valeur de 0,1916 dans le pire des cas après la calibration, ce qui est une nette amélioration et correspond aux résultats obtenus précédemment lors d'un essai de calibration avec uniquement les

TIV de suivi inclus dans la fonction objectif, et dont les résultats se retrouvent dans un article à paraître (Gauthier, Saunier, Le Digabel, & Cao, 2016).

Le second résultat est quant à lui moins encourageant : la distribution simulée des TIV acceptés s'est beaucoup moins améliorée durant le processus de calibration, avec une valeur de la fonction objectif finale plus proche de celle de départ. Globalement, donc, NOMAD n'a pas réussi à obtenir de points fournissant une aussi bonne amélioration par rapport au point initial en utilisant cette fonction objectif. Or, il semble que l'utilisation de cet indicateur à l'intérieur de la fonction objectif ait freiné la capacité de NOMAD à améliorer les TIV acceptés, ce qui rend le résultat final moins satisfaisant que celui présenté dans (Gauthier et al., 2016). L'information est reprise dans la Figure 5-2 et dans la Figure 5-3, qui montrent respectivement les distributions cumulées des TIV de suivi et des TIV acceptés pour le point initial, le point calibré et pour les observations terrains à reproduire. On remarque que dans tous les cas, la distribution calibrée est légèrement plus collée sur la distribution observée, mais cette amélioration est surtout visible dans les petits TIV de suivi, alors que les TIV de suivi supérieurs à 2 secondes tendent à ne pas être mieux distribués après la calibration qu'avant. Dans le cas des TIV acceptés, aucune amélioration notable n'est visible, corroborant les conclusions tirées du Tableau 5-2.

La Figure 5-4 permet de visualiser la progression des différentes composantes de la fonction objectif globale durant le processus de calibration. Dans cette figure, seuls les points respectant toutes les contraintes de calibration, et donc considérés dans le processus de détection du meilleur point global, sont illustrés. Le meilleur point lors de cette calibration a été obtenu à l'essai 73, ce qui explique que rapidement NOMAD se soit mis à réduire sa discréétisation de l'espace afin d'atteindre son critère de convergence et que les résultats soient à peu près stables à partir de l'évaluation 250. Il apparaît que le critère de convergence donné à NOMAD soit trop petit pour les besoins d'une calibration de VISSIM puisque des centaines d'évaluations n'ont pas suffi à l'atteindre. Il serait important d'étudier comment mieux paramétrier NOMAD de sauver des évaluations lors de futures études utilisant la méthode proposée.

On note que les résultats, même après le meilleur point, sont très bruités, et ceci particulièrement dans le cas des TIV acceptés. Ceci s'explique par les processus stochastiques de la simulation, qui ont pour effet que d'une série de réplications à une autre, même avec des paramètres presque semblables comme c'est le cas lors de la convergence de l'algorithme, les résultats peuvent varier

légèrement à la hausse ou à la baisse. Dans les premiers essais, on remarque que certains des TIV de suivi sont le facteur limitant de la calibration, puis que NOMAD trouve rapidement un jeu de paramètres qui améliore l'adéquation de ces distributions. À ce point, les TIV acceptés deviennent le facteur limitant. On remarque également que les TIV de suivi ont parfois le pire résultat au test de Kolmogorov-Smirnov pour certain jeu de paramètres, notamment avec ceux de la simulation des données de la vidéo A13-13 qui présente des pics importants. Dans l'ensemble toutefois, NOMAD n'arrive plus à améliorer la fonction objectif puisque les TIV acceptés ne semblent pouvoir être améliorés et qu'après le creux du 73<sup>e</sup> point, aucune amélioration n'est notée.

Dans l'ensemble, il apparaît donc que les résultats obtenus pour les TIV de suivi sont acceptables, mais ne sont pas optimaux puisque leur amélioration a été limitée par le fait que les TIV acceptés n'ont jamais connu d'amélioration, peu importe les essais effectués par NOMAD. La Figure 5-1 montre également, plus de 77 % des points testés sont des points réalisables alors que cette proportion chute à 25 % pour les 150 premiers points testés. La recherche dans l'espace est donc concentrée au tout début du processus. Une analyse de l'historique de calibration montre d'ailleurs que la variation du nombre de changements de voies opportunistes effectués dans les simulations réalisables est de moins de 5 % par rapport à la simulation initiale. Il apparaît donc que le problème principal soit le fait que les jeux de paramètres testés ne semblent pas influencer ce comportement. Son inclusion dans la fonction objectif était donc une erreur.

Des essais de calibration par poids selon les différents indicateurs permettraient possiblement de capturer une certaine amélioration des TIV de suivi alors que les TIV acceptés stagnent. Or, cette approche a comme défaut qu'une amélioration d'une des composantes de la fonction objectif peut cacher la détérioration d'une autre, tant que la somme des deux continue de diminuer. Le choix le plus judicieux, à ce point, serait donc de tenter une calibration des TIV acceptés uniquement et de rechercher s'il est, dans un premier temps, possible de les calibrer et si oui, avec quel indicateur et quelle fonction d'ajustement. Dans un deuxième temps, il serait important de tester différentes versions de la fonction objectif afin de voir s'il est possible d'effectuer un arbitrage moins restreignant entre l'amélioration de la portion des TIV de suivi et l'amélioration de la portion des TIV acceptés. Une autre piste de solution pourrait être de tenter une calibration bi-objective afin que les deux portions de la fonction soient traitées indépendamment, permettant d'améliorer les TIV de suivi sans égard à l'amélioration observée sur les TIV acceptés. Cette solution n'est

cependant valide que si des paramètres ayant un effet sur les TIV acceptés sont identifiés préalablement, sinon leur calibration restera un échec.

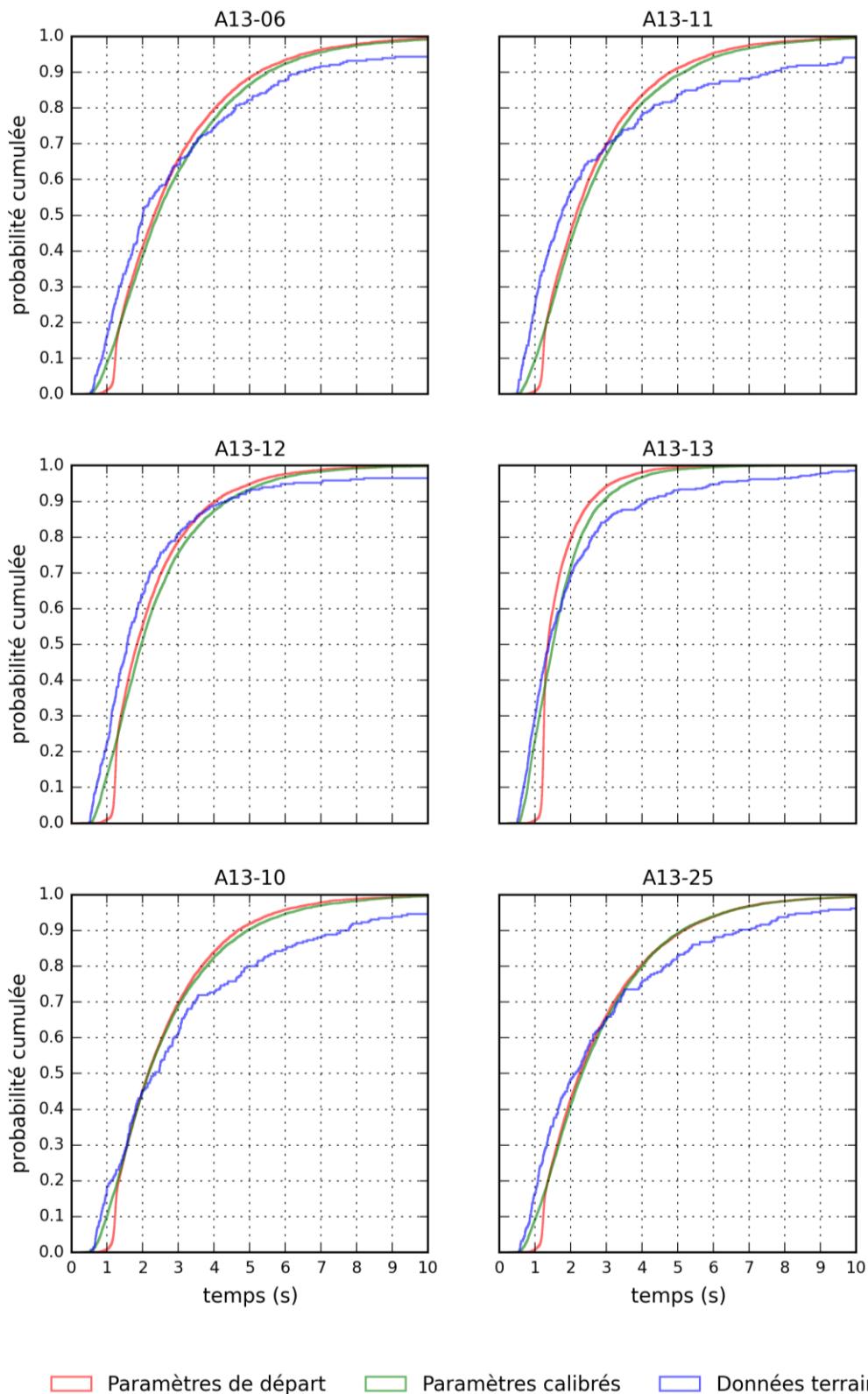


Figure 5-2 : Distributions cumulées des TIV de suivi pour les vidéos utilisées lors des étapes de calibration (A13-06, A13-11, A13-12 et A13-13) et de validation (A13-10 et A13-25)

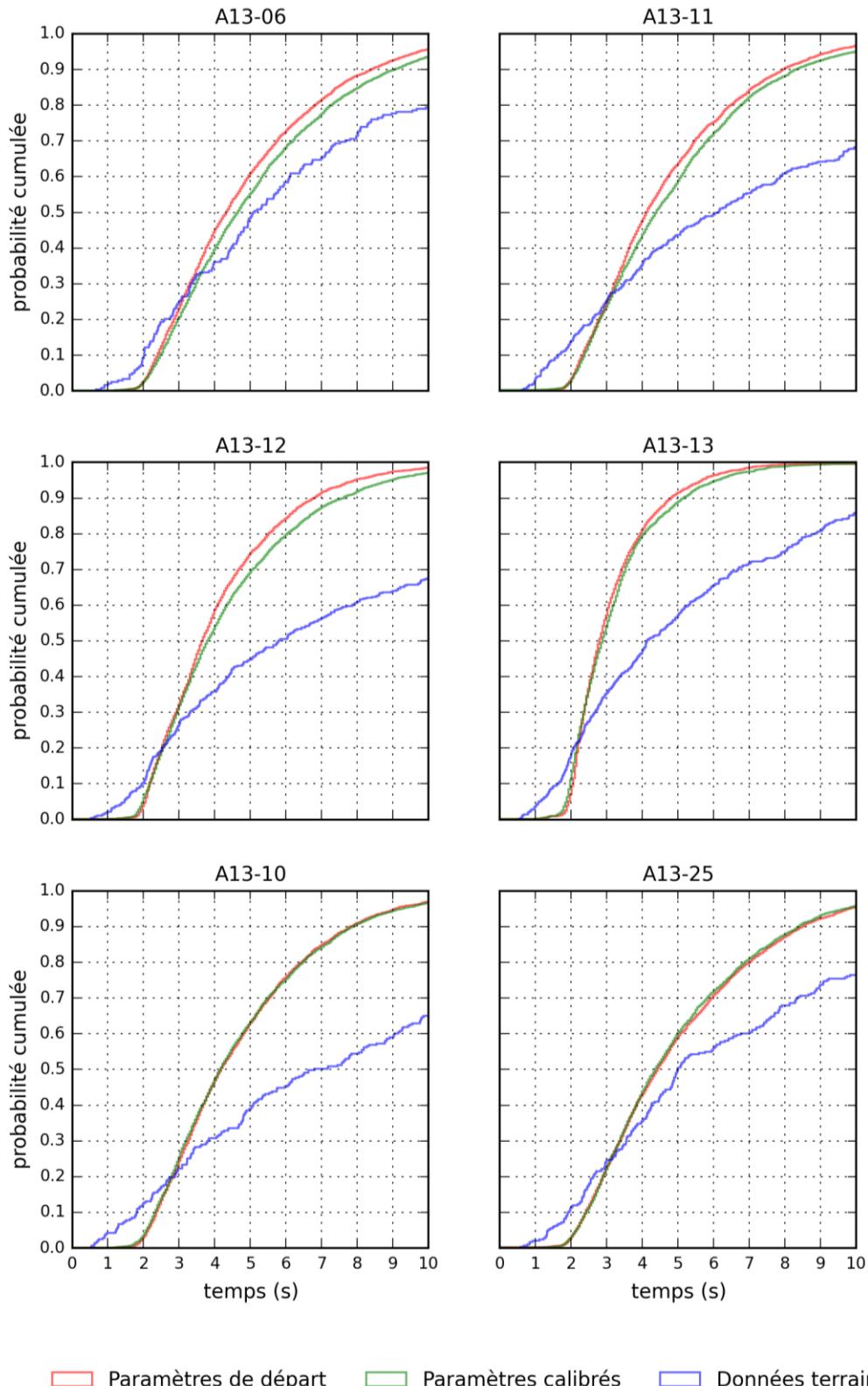
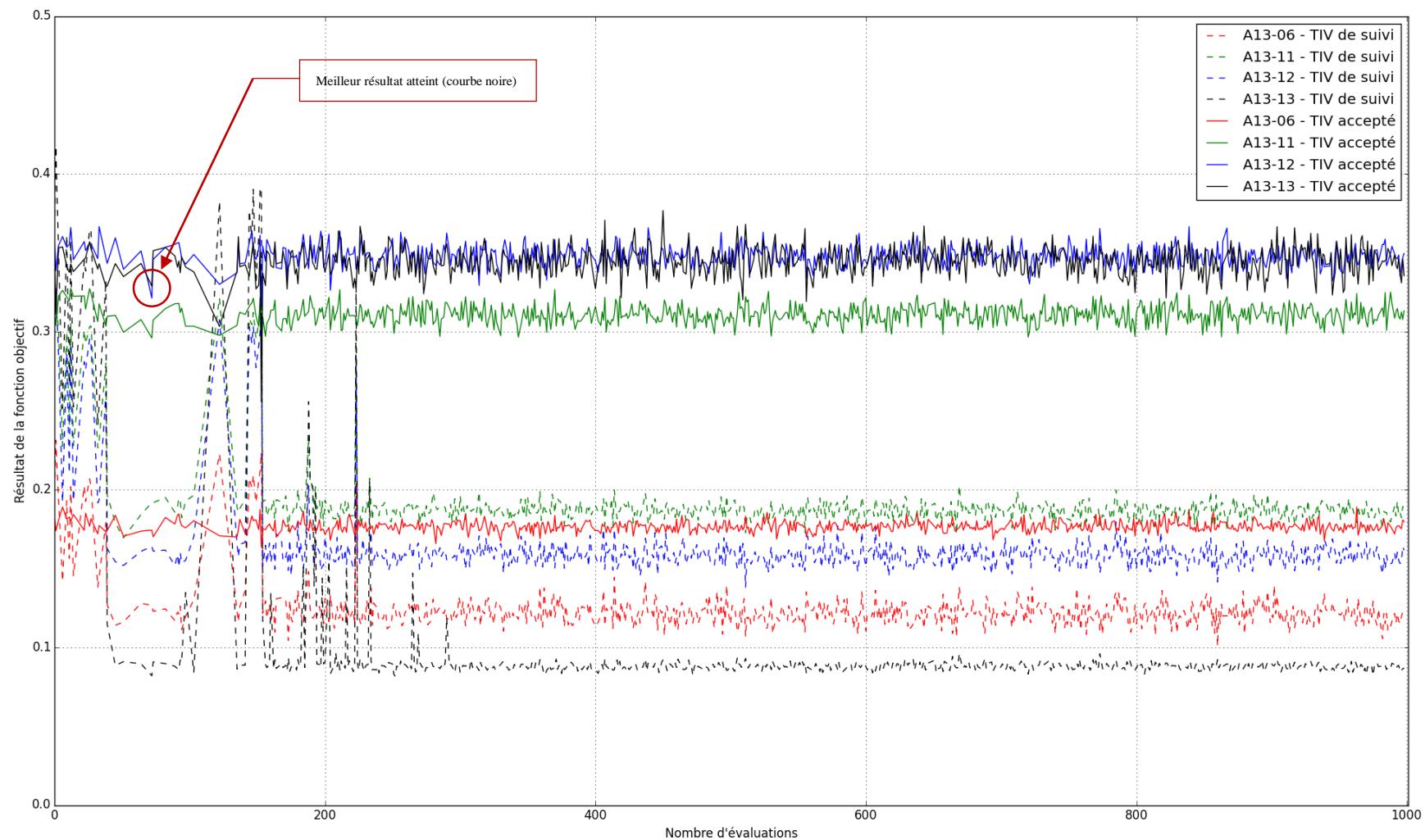


Figure 5-3 : Distributions cumulées des TIV acceptés pour les vidéos utilisées lors des étapes de calibration (A13-06, A13-11, A13-12 et A13-13) et de validation (A13-10 et A13-25)



*Seuls les points réalisables sont tracés sur cette figure, ce qui explique que la variation semble moins importante dans certaines régions*

Figure 5-4 : Visualisation de la progression des différentes composantes de la fonction objectif en fonction du nombre de jeux de paramètres évalués par NOMAD

## 5.2 Paramètres calibrés à utiliser

L’application de cette méthode avait mené dans un premier temps à la soumission pour publication d’un article (Gauthier et al., 2016) utilisant une première version de la méthode présentée dans ce mémoire. Or, il apparaît que les résultats obtenus avec cette première version sont plus satisfaisants que ceux présentés avec la méthode complète telle que présentée dans le Tableau 5-1. Par conséquent, la présente section de recommandation effectue un rappel des hypothèses de travail utilisées dans le cadre de cet article avant de présenter les paramètres recommandés.

Tableau 5-3 : Paramètres des modèles Wiedemann 99 et de changement de voie proposés pour les segments autoroutiers en section droite de la région montréalaise.

<i>Modèle de poursuite</i>	<i>Paramètre</i>	<i>Valeur par défaut</i>	<i>Valeur calibrée</i>
<i>Modèle de changement de voie</i>	w99cc0	1,5	0,894
	w99cc1	0,9	0,006
	w99cc2	4,0	0,671
	w99cc3	-8,0	-6,658
	w99cc4	-0,35	0,992
	w99cc5	0,35	0,350
	w99cc6	11,44	17,925
	w99cc7	0,25	-5,117
	w99cc8	3,5	5,462
	w99cc9	1,5	120,00
	ObsrvdVeh	2	8
<i>Modèle de changement de voie</i>	MaxDecelOwn	-4,0	-3,337
	DecelRedDistOwn	100,0	104,472
	AccDecelOwn	-1,0	-1,000
	MaxDecelTrail	-3,0	-3,446
	DecelRedDistTrail	100,0	91,056
	AccDecelTrail	-1,0	-9,397
	DiffusTm	60,0	158,387
	MinHdwy	0,5	8,550
	SafDistFactLnChg	0,6	0,578

Tableau tiré de (Gauthier et al., 2016)

Tel que discuté dans la section précédente, le processus de calibration présenté dans ce mémoire n’a pas été en mesure d’effectuer l’optimisation en regard des TIV de suivi puisque leur amélioration était occulté par la stagnation des indicateurs ayant trait aux TIV acceptés. Ainsi, NOMAD ne recevait pas l’information à l’effet qu’un des modèles pouvait être amélioré. Il lui était alors impossible de sélectionner comme meilleur point un jeu de paramètres donnant de meilleurs

résultats sur les TIV de suivi uniquement. Or, cet exercice avait été effectué auparavant, et nous reprenons donc ici les conclusions de cette calibration.

Rappelons tout d'abord les hypothèses de travail de cette calibration, qui diffèrent très peu de celles présentées au Chapitre 3. Les données utilisées pour les phases de calibration et de validation sont les mêmes et les paramètres choisis pour la calibration exactement les mêmes. De plus, le point de départ, les contraintes de calibration et le budget d'évaluation étaient identiques et l'étude concernant le nombre de répétitions à effectuer avait également donné 20 comme résultat. La seule différence résidait en fait dans la formulation de la fonction objectif puisque les indicateurs concernant les TIV acceptés n'en faisaient pas partie. La fonction objectif était donc de la même forme que celle présentée à la section 3.3.1 sans la portion concernant  $f_2$ . Les résultats obtenus lors de cette calibration sont reproduits dans le Tableau 5-4 et les valeurs des paramètres correspondant à ces résultats sont compilées dans le Tableau 5-3.

Tableau 5-4 : Résultat de la fonction objectif pour les paramètres proposés.

<i>Vidéo*</i>	<i>Débit</i> (veh/h/ln)	<i>Statistique D du test de Kolmogorov-Smirnov</i>	
		<i>Valeurs par défaut</i>	<i>Valeurs calibrées</i>
<b>Calibration</b>			
A13-06	1252	0.2025	0.0933
A13-11	1352	0.3069	0.1447
A13-12	1556	0.2961	0.1311
A13-13	2140	0.3576	0.0879
<i>Fonction objectif</i>	--	0.3069	0.1447
<b>Validation</b>			
A13-10	1352	0.185	0.1181
A13-25	1296	0.224	0.0972

Tableau tiré de (Gauthier et al., 2016)

En analysant les résultats du Tableau 5-4, on remarque que la calibration a été beaucoup plus efficace avec des valeurs de la fonction objectif en deçà de celles trouvées pour les TIV de suivi dans le Tableau 5-2 pour toutes les vidéos, à l'exception de la vidéos A13-13 qui obtient essentiellement le même résultat dans les deux calibrations. Tel que discuté dans la section 5.1, la portion de la fonction objectif traitant des TIV acceptés bloquait la progression de l'algorithme d'optimisation puisqu'aucune amélioration n'a pu être trouvée pour cet indicateur. Comme

l'algorithme ne recevait pas d'information sur l'amélioration de l'ajustement des TIV de suivi, les points testés ne cherchaient pas à améliorer cet indicateur, mais bien uniquement les TIV acceptés. Dans (Gauthier et al., 2016), la valeur de fonction objectif présentait une amélioration au fur et à mesure de la progression de la calibration, ce qui explique que l'algorithme ait pu tester des points permettant d'observer un meilleur résultat pour les TIV de suivi.

Tel que mentionné plus haut, les TIV acceptés n'avaient pas été calculés lors de l'essai de calibration effectué dans (Gauthier et al., 2016), ce qui explique que l'algorithme n'est pas été bloqué et ait réussi à trouver de meilleurs résultats pour la calibration des TIV de suivi. Or, le fait de ne pas avoir calibré les TIV acceptés n'est a priori pas un problème puisque, comme on le constate dans la Figure 5-4 et il a été discuté à la section 5.1, toutes les combinaisons de paramètres testés ont donné sensiblement le même résultat sur cet indicateur. Il n'y a donc pas lieu de croire que la capacité du modèle à reproduire les TIV acceptés soit meilleure ou pire dans le cas du point calibré obtenu dans l'article. En conséquence, les valeurs des paramètres rapportés au Tableau 5-3 sont recommandés pour les conditions de simulations suivantes : section droite d'autoroute avec un débit entre 1250 et 2150 veh/h/ln et utilisant les informations de bases fournies contenu dans le fichier VISSIM de base de la Ville de Montréal.

## CHAPITRE 6 CONCLUSION ET RECOMMANDATIONS

### 6.1 Conclusion générale

La recherche présentée dans ce document visait le développement d'une procédure de calibration des logiciels de microsimulation routière dans l'optique d'augmenter leur capacité à reproduire les conditions routières d'une zone donnée. Pour y arriver, un logiciel dont la fonction est de permettre la communication entre un algorithme d'optimisation et le logiciel à calibrer ont été développés et la méthode a été testée et validée sur un projet pilote consistant à calibrer les comportements de conduite du logiciel VISSIM pour les autoroutes montréalaises. Pour être plus précis, c'est la combinaison des modèles de poursuite Wiedemann 99 et de changements de voie qui a été calibrée.

La méthode présentée s'appuie sur des outils ouverts, disponibles gratuitement sur les sites des chercheurs les ayant développés. Ces outils sont Traffic Intelligence, un algorithme de détection et suivi des véhicules dans des images vidéo, et NOMAD, un logiciel d'optimisation sans dérivées implantant l'algorithme du Mesh Adaptive Direct Search. Le logiciel développé (dans le langage Python) permet la communication entre NOMAD et le logiciel à calibrer afin de tester un jeu de paramètres donnés puis convertit les extrants de ce logiciel pour les comparer aux données terrain afin de transmettre à NOMAD l'information dont il a besoin pour continuer le processus d'optimisation, soit le résultat de la fonction objectif et le respect des contraintes par ce jeu de paramètres.

L'étude de cas présente également une technique de distinction des types de virages afin de classifier les différents changements de voies détectés dans la simulation ou dans des données vidéos selon qu'ils sont effectués par le conducteur pour atteindre une destination ou s'ils sont effectués afin de conserver ou d'atteindre une vitesse désirée sans être obligatoire pour atteindre une destination particulière. Cette technique est utilisée afin de ne comptabiliser que les changements de voies classées opportunistes dans l'étude, puisque ceux-ci ne sont pas automatiquement effectués selon le modèle de VISSIM alors que les changements de voies obligatoires menant au respect de la trajectoire globale du véhicule sont automatiquement effectués.

Finalement, une technique de détermination du nombre de répétitions à effectuer est discutée et utilisée, puisque la méthode généralement utilisée se basant sur la moyenne d'une distribution

normale ne peut s'appliquer lorsque la fonction objectif repose sur des mesures d'adéquation des distributions des indicateurs observés et simulés.

## 6.2 Limitations de la méthode et pistes de solutions

Les limitations peuvent être regroupées en deux grands groupes, soit les limites d'utilisation, essentiellement techniques, de la méthode globale et les sources d'erreurs potentielles sur les paramètres recommandés à la suite de l'implantation de celle-ci dans le cadre du projet de calibration de VISSIM pour les autoroutes montréalaises.

### 6.2.1 Limitations techniques

Le schéma méthodologique présenté à la section 3.1.1 se veut généralisable à tous les logiciels de microsimulation et à un grand nombre d'indicateurs et de données, bien que nous ne démontrions son applicabilité que sur un cas spécifique et un seul logiciel. Par contre, sa rapidité d'exécution est directement dépendante de la taille du réseau simulé et du nombre de véhicules présents dans ce réseau et un autre projet en cours démontre que des problèmes de traitement des résultats de la simulation émergent lorsque le réseau est substantiellement plus grand que celui utilisé dans la présente étude.

En effet, plus le nombre de véhicules dans le réseau est important et plus les calculs effectués par le logiciel sont nombreux, affectant du même coup le temps nécessaire pour simuler et traiter chaque réPLICATION. De plus, la méthode actuellement implantée reprend une portion des calculs effectués dans le logiciel afin de calculer les différents indicateurs utilisés et le résultat des contraintes et de la fonction objectif. Par exemple dans le cas d'étude ci-haut, les TIV sont recalculés par le programme Python alors que le logiciel les connaît déjà lors de la simulation. La nouvelle version de VISSIM (8.0) permet d'ailleurs de calculer certain indicateurs lors de la simulation et d'autres logiciels permettent également d'effectuer ce genre de tâches. Il serait judicieux de migrer certains des calculs dans ces outils afin de réduire le temps total nécessaire pour simuler et traiter les résultats de chaque réPLICATION, puisque bien que le temps additionnel de calcul n'est pas problématique à la bonne marche de la méthode, il peut rapidement devenir irréaliste d'attendre un résultat plusieurs semaines. D'autant plus si plusieurs essais de calibration sont menés afin de pouvoir apporter des modifications, par exemple pour ajuster des contraintes de

calibration ou encore séparer différemment les données entre les processus de calibration et de validation selon les résultats d'un premier essai.

La façon la plus facile de pallier ce problème est de faire ce qu'il a été décidé de faire dans le cadre du projet de calibration de VISSIM pour les autoroutes montréalaises, soit de travailler sur un petit tronçon représentatif pour effectuer la calibration et d'ensuite valider les résultats sur un plus large réseau, pour voir si la calibration effectuée à petite échelle est effectivement transférable, ou si des contraintes supplémentaires doivent être apportées afin de mieux représenter les conditions du grand réseau dans le petit lors d'un futur essai de calibration.

Or, cette solution n'est pas viable pour tous les modèles à calibrer, puisque ceux-ci peuvent n'avoir de signification que sur le modèle complet. On peut penser entre autres aux modèles d'affectations de chemin entre les différents points d'une matrice origine-destination qui peuvent difficilement être rapportés à une échelle inférieure sans perdre des degrés de liberté indispensables à leur fonctionnement. Dans ces cas, la solution passe par le développement d'une meilleure communication entre la méthode de calibration et le logiciel calibré et par une meilleure parallélisation des différentes étapes de calcul. Améliorer la communication permettrait de ne pas dédoubler certains calculs déjà effectués par le logiciel à calibrer et d'ainsi réduire le temps additionnel que prend la méthode une fois la simulation effectuée. Une meilleure parallélisation passe entre autres par le fait de distribuer les simulations des différentes répétitions nécessaires sur plus d'un processeur à la fois, possiblement sur des machines distinctes, puisque pour des tailles plus grandes de réseau, c'est cette portion de la méthode qui devient le point critique en termes de temps. Par contre, cette solution peut être difficilement implantable dans certains cas puisque plusieurs des logiciels de microsimulation sont sujets à des limites sur le nombre d'instances par licence détenue par l'utilisateur. Dans le cas de notre projet, nous utilisions déjà les 4 instances disponibles à la fois et il aurait alors fallu acheter une autre licence afin d'implémenter cette solution.

### **6.2.2 Sources d'erreurs sur le calcul des paramètres**

Les sources d'erreur sur les valeurs des paramètres peuvent être résumées en un facteur : la faible disponibilité de conditions de circulation permettant de vérifier si les paramètres proposés sont valides dans une grande plage de débits, tant en régime fluide qu'en congestion routière. La conséquence première est qu'il est impossible de prédire si les paramètres en questions permettront

d'obtenir des modélisations réalistes, ou même utilisables, pour des conditions différentes, notamment puisque le fait de modifier les paramètres de conduite engendre parfois des simulations dont les conditions routières sont carrément irréalistes, comme le montre la Figure 4-9. De plus, il est possible de voir apparaître des conditions routières irréalistes dans une simulation alors qu'une autre simulation utilisant les mêmes paramètres, mais ayant un débit d'entrée différent présentait des conditions de circulation tout à fait acceptables.

Il est toutefois possible de nommer plusieurs causes à cette faible diversité à l'intérieur des données, tel que présenté dans les sous-sections suivantes.

#### **6.2.2.1 Meilleur positionnement de la caméra**

Lors de la collecte de données, il peut parfois être complexe de trouver une combinaison de sections de route intéressantes et qui présentent également une infrastructure facilement accessible afin d'installer le mat de collecte vidéo avec un champ adapté à la collecte de données de circulation. Or, comme le discutent (Anderson-Trocmé et al., 2015), le positionnement de la caméra a un grand effet sur les positions perçues des véhicules, allant jusqu'à créer des surestimations des vitesses de l'ordre de 14 %. Même si la vitesse n'est pas directement utilisée dans la méthodologie présentée, elle est tout de même dérivée des positions, qui elles sont utilisées pour calculer les temps intervéhiculaires. Or, comme discuté à la section 3.2.4.1, l'erreur de parallaxe augmente sur les plus gros véhicules lorsque ceux-ci sont suivis loin du sol, pouvant carrément mener à leur détection dans une voie de circulation voisine de la voie qu'ils empruntent réellement, et amenant l'algorithme à calculer une interaction inexistante en réalité avec le véhicule qui semble alors le suivre ou le précéder.



Figure 6-1 : Exemple de camions dans la voie du centre de l'A15 masquant la vision d'une autre voie de circulation.

De plus, le positionnement de la caméra peut avoir une influence sur le nombre de véhicules détectés. Le cas de la collecte sur l'A15 soulignait à grands traits ce problème, puisque les véhicules lourds sont nombreux, au lieu de collecte, à se déplacer dans la voie du centre, probablement pour éviter la zone de convergence entre la voie d'accélération et la voie de droite et parce que la voie de gauche leur est interdite à partir d'une dizaine de mètres en aval. Avec le positionnement de la caméra, toutefois, l'effet direct est que ces camions cachent entièrement la voie de droite ainsi que la portion de la voie d'accélération visible dans le champ de la caméra, tel qu'on peut le voir à la Figure 6-1. Par conséquent, en congestion, le nombre de véhicules non détectés, car tout simplement non visibles est élevé et les interactions visibles dans cette première voie lorsqu'il n'y a momentanément pas de camion au centre ne sont pas représentatives de l'ensemble des interactions. Un positionnement de la caméra sur le viaduc situé non loin et faisant directement face aux voies de circulation aurait grandement aidé sur ce point, quitte à devoir surveiller l'équipement de collecte durant les dix heures de filmage faute de pouvoir installer la caméra de sorte qu'il soit impossible de la voler comme c'est le cas lorsqu'elle est entièrement déployée en haut du mat. L'utilisation des bâtiments avoisinants aurait également pu être une solution, puisque

la vue alors encore plus plongeante aurait permis d'éliminer en partie le masquage occasionné par les véhicules lourds.

### **6.2.2.2 Calibration de Traffic intelligence**

Le positionnement de la caméra de collecte ne règle cependant pas tous les problèmes liés à la collecte de données. Même pour des véhicules bien visibles et une erreur de parallaxe sans conséquences, il existe des situations où l'algorithme de détection des véhicules n'est pas performant dans son état actuel. La plus grande difficulté apparaît dans des conditions congestionnées, où les véhicules sont très rapprochés les uns des autres et où les véhicules effectuent souvent des arrêts presque complets avant de redémarrer pour quelques mètres. Ces conditions mettent aux défis la reconnaissance du mouvement et le groupement des différents pixels, traduits en trajectoires de points particuliers et en des objets distincts. Des travaux de calibration de l'algorithme de détection ont d'ailleurs été effectués pour le mouvement piétonnier, dont l'une des caractéristiques est justement un groupement des individus qu'il faut ensuite distinguer les uns des autres (Ettehadieh, 2014). Une technique similaire à celle utilisée pour la calibration des paramètres pour le suivi piétonnier pourrait permettre d'améliorer la performance de l'algorithme de détection et suivi de Traffic intelligence et donc d'augmenter le nombre de vidéos dont les résultats peuvent être utilisés. D'ailleurs ces travaux ont été débutés, notamment par (Morse, St-Aubin, Miranda-Moreno, & Saunier, 2016).

### **6.2.2.3 Collecte sur une plus grande période de temps**

Finalement, une collecte vidéo effectuée sur une plus grande période de temps permettrait de capturer les variations journalières de comportement routier qui fait en sorte qu'une même route peut être congestionnée une certaine journée et relativement plus fluide le lendemain à la même heure. Avoir une plus grande plage de données aux différentes heures permettrait également d'étudier le comportement routier à différents moments de la journée pour déterminer si des paramètres différents doivent être utilisés pour les pointes du matin et du soir et pour la période entre les pointes, ou si au contraire un seul jeu de paramètre est suffisant pour décrire l'ensemble de la journée. Malheureusement, vu le nombre restreint de données disponibles, une telle étude n'a pas pu être effectuée et les données des différentes périodes de la journée ont été considérées ensemble afin d'obtenir suffisamment de données utilisables pour appliquer la méthode de calibration.

### 6.3 Perspectives de recherche

Les sections précédentes ont déjà fait état de quelques pistes de recherche et de développement qui permettraient d'améliorer les résultats futurs d'études semblables : une attention particulière au positionnement de la caméra lors de la collecte, la calibration de Traffic Intelligence pour le suivi des véhicules en situation congestionnée, ainsi que la collecte de données sur une plus grande période. En plus de ces actions, qui permettraient d'améliorer la qualité et la portée des résultats présentés dans cette étude, il est possible de développer les outils présentés afin d'agrandir la portée réelle et l'applicabilité de la méthode présentée.

Tel que discuté tout au long de ce mémoire, CATS ne peut à l'heure actuelle implémenter la méthode proposée que sur un nombre restreint d'indicateurs et uniquement pour le logiciel VISSIM. La première étape de généralisation serait donc d'étendre son champ d'action à d'autres logiciels, ce qui demande tout de même une bonne adaptation des fonctionnalités puisque les modules de communications externes ne sont pas identiques d'un logiciel à l'autre. De même, les différents logiciels de microsimulation n'implémentent pas tous les mêmes modèles de description de la circulation ou encore ne permettent pas de jouer avec le même nombre de paramètres. Cette généralisation est donc une tâche fastidieuse, puisqu'elle implique également une bonne compréhension des logiciels et de la façon de les programmer, mais elle constitue une étape essentielle pour toute tentative de recherche les utilisant.

Ensuite, il serait important de varier les sources de données possibles que CATS peut accepter, afin de permettre de calibrer des modèles parfois plus simples, mais plus couramment utilisés comme la distribution des arrivées pour un réseau donné. Dans le même ordre d'idée, il est nécessaire d'augmenter le nombre d'indicateurs et de fonctions d'ajustement que CATS peut calculer, afin justement de pouvoir traiter ces nouvelles sources de données adéquatement.

Ces divers ajouts permettront par la suite de poser des questions de recherche plus complexes, puisqu'un outil pour les traiter aura déjà été mis sur pied. Parmi ces questions, on retrouve entre autres ce qui intéresse un nombre grandissant d'ingénieurs : quelle est la performance relative des modèles microscopiques et des modèles mésoscopiques, soit des modèles avec moins de détails pour décrire la circulation, surtout au vue du type de données disponibles pour effectuer la calibration des modèles et l'évaluation des différents scénarios? Quels types d'indicateurs et d'impacts sont mieux représentés par des modélisations sur ces deux familles de modèles? Y'a-t-

il un avantage à continuer d'utiliser les modèles microscopiques sachant qu'ils demandent des calculs très coûteux?

Même sans entrer dans les questions de recherche nouvelles, on peut aisément voir l'applicabilité de la présente méthode pour la calibration des réseaux microscopiques. En effet, la portée réelle des paramètres qui sont fournis dans ce mémoire est assez limitée, puisqu'elle ne concerne que les sections droites d'autoroutes. Il est possible de réutiliser cette méthode directement, ou avec les ajouts permettant de calibrer d'autres logiciels, afin de bonifier les réseaux décrivant la région montréalaise sur d'autres types de routes. Ces autres types de routes sont variés, comme les sections d'entrecroisement autoroutières qui devaient être abordées par la présente recherche, les sections artérielles urbaines à haute densité, comme au centre-ville, ou à basse densité, les sections résidentielles ou encore avec forte présence de camionnage comme aux abords du Port de Montréal ou dans les secteurs industriels. De même, la méthodologie pourrait être reprise afin de calibrer d'autres réseaux au Québec, par exemple ceux de la Capitale nationale ou encore de la région de Saguenay, pour tous les types de routes. Les possibilités ouvertes par une telle méthode sont nombreuses, et ce mémoire présente des pistes de solution afin d'éviter les erreurs liées à la collecte de données et pour tester d'autres indicateurs et d'autres types de fonctions objectifs.

## BIBLIOGRAPHIE

- Aghabayk, K., Sarvi, M., Young, W., & Kautzsch, L. (2013, 2013). *A novel methodology for evolutionary calibration of vissim by multi-threading*. Paper presented at the 36th Australasian Transport Research Forum, Brisbane, Australia.
- Anderson-Trocmé, P., Stipancic, J., Miranda-Moreno, L., & Saunier, N. (2015). *Performance Evaluation and Error Segregation of Video-Collected Traffic Speed Data*. In Transportation Research Board Annual Meeting Compendium of Papers, (pp. 15-1337).
- Antoniou, C., Barcelò, J., Brackstone, M., Celikoglu, H., Ciuffo, B., Punzo, V., Sykes, P., Toledo, T., Vortisch, P., & Wagner, P. (2014). *Traffic Simulation: Case for guidelines*. Luxembourg: Publications Office of the European Union, JRC Scientific and Technical Research Reports.
- Audet, C. (2014). *A survey on direct search methods for blackbox optimization and their applications*, in Mathematics Without Boundaries, Surveys in Interdisciplinary Research (pp. 31-56). New York: Springer.
- Audet, C., Béchard, V., & Le Digabel, S. (2008). Nonsmooth Optimization through Mesh Adaptive Direct Search and Variable Neighborhood Search. *Journal of Global Optimization*, vol. Vol. 41 (2), pp. 299-318
- Audet, C., & Dennis, Jr., J. E. (2006). Mesh Adaptive Direct Search Algorithms for Constrained Optimization. *SIAM Journal on Optimization*, vol. 17 (1), pp. 188-217
- Audet, C., & Dennis, J., J E. (2009). A progressive barrier for derivative-free nonlinear programming. *SIAM Journal on Optimization*, vol. 20 (1), pp. 445-472
- Audet, C., Ianni, A., Le Digabel, S., & Tribes, C. (2014). Reducing the Number of Function Evaluations in Mesh Adaptive Direct Search Algorithms. *SIAM Journal on Optimization*, vol. Vol. 24 (5), pp. 621-642
- Chatterjee, I., Edara, P., Menneni, S., & Sun, C. (2009). Replication of work zone capacity values in a simulation model. *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2130, pp. 138-148

- Chu, L., Liu, H. X., Oh, j.-S., & Recker, W. (2003). *A Calibration Procedure for Microscopic Traffic Simulation*. Paper presented at the IEEE Intelligent transportation Systems.
- Ciuffo, B., Punzo, V., & Montanino, M. (2012). *The Calibration of Traffic Simulation Models : Report on the assessment of different Goodness of Fit measures and Optimization Algorithms MULTITUDE Project – COST Action TU0903*. Luxembourg: Publications Office of the European Union, JRC Scientific and Technical Research Reports.
- Conn, A. R., Scheinberg, K., & Vincente, L. N. (2009). *Introduction to Derivative-Free Optimization*. Philadelphia, USA: Society for Industrial and Applied Mathematics and the Mathematical Programming Society.
- Eberhart, R., & Kennedy, J. (1995, 4-6 Oct 1995). *A new optimizer using particle swarm theory*. Paper presented at the Micro Machine and Human Science, 1995. MHS '95., Proceedings of the Sixth International Symposium on.
- Eiben, A. E., Raué, P. E., & Ruttkay, Z. (1994). *Genetic algorithms with multi-parent recombination*. In Y. Davidor, H.-P. Schwefel, & R. Männer, in Parallel Problem Solving from Nature — PPSN III (Vol. 866, pp. 78-87): Springer Berlin Heidelberg.
- Ettehadieh, D. (2014). *Systematic Parameter optimization and Application of Automated tracking in Pedestrian-Dominant Situations*. (Maîtrise ès Sciences Appliquées), École Polytechnique de Montréal, Montréal.
- FHWA. (2004a). Traffic Analysis Toolbox Volume II: Decision Support Methodology for Selecting Traffic Analysis Tools. McLean, VA, USA: Federal Highway Administration.
- FHWA. (2004b). Traffic Analysis Toolbox Volume III: Guidelines for Applying Traffic Microsimulation Modeling Software. McLean, VA, USA: Federal Highway Administration.
- Gamboa, K. (2014). *Analyse de sensibilité pour la calibration et validation du réseau autoroutier congestionné de l'île de Montréal avec VISSIM*. (Maîtrise professionnelle), École Polytechnique de Montréal, Montréal.
- Gauthier, L., Saunier, N., Le Digabel, S., & Cao, G. (2016). *Calibration of Driving Behavior Models using Derivative-Free Optimization and Video Data for Montreal highways*, in Transportation Research Board Annual Meeting Compendium of Papers 2016.

- Hellinga, B. R. (1998). Requirements for the calibration of traffic simulation models. *Proceedings of the Canadian Society for Civil Engineering*, vol. 4, pp. 211-222
- Hollander, Y., & Liu, R. (2008a). Estimation of the distribution of travel times by repeated simulation. *Transportation Research Part C: Emerging Technologies*, vol. 16 (2), pp. 212-231
- Hollander, Y., & Liu, R. (2008b). The principles of calibrating traffic microsimulation models. *Transportation*, vol. 35 (3), pp. 347-362
- Jackson, S., Miranda-Moreno, L., St-Aubin, P., & Saunier, N. (2012). A Flexible, Mobile Video Camera System and Open Source Video Analysis Software for Road Safety and Behavioural Analysis. *Transportation Research Record: Journal of the Transportation Research Board*, vol. Vol. 2365, pp. 90-98
- Jie, L., Zuylen, H. V., Chen, Y., Viti, F., & Wilmink, I. (2013). Calibration of a microscopic simulation model for emission calculation. *Transportation Research Part C*, vol. 31 (2013), pp. 172-184
- Jodoin, J. P., Bilodeau, G. A., & Saunier, N. (2014, 24-26 March 2014). *Urban Tracker: Multiple object tracking in urban mixed traffic*. Paper presented at the Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on.
- Kim, K.-O., & Rilett, L. (2003). Simplex-based calibration of traffic microsimulation models with intelligent transportation systems data. *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1855 (1), pp. 80-89
- Kim, S.-J. K., Wonho, & Rilett, L. R. (2005). Calibration of Microsimulation Models Using Nonparametric Statistical Techniques. *Transportation Research Record: Journal of the Transportation Research Board*, vol. Vol. 1935 (2005 Information Systems and Technology), pp. 111-119
- Kolda, T. G., Lewis, R. M., & Torczon, V. (2003). Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods. *SIAM Review*, vol. 45 (3),
- Le Digabel, S. (2011). Algorithm 909: NOMAD: Nonlinear Optimization with the MADS Algorithm. *ACM Transactions on Mathematical Software*, vol. 37 (4), pp. 44:41-44:15

- Li, Z., Liu, H., & Li, J. (2010). *A Calibration and Validation Procedure for Microscopic Simulation Model*. Paper presented at the 13th International IEEE Annual Conference on Intelligent Transportation Systems, Madeira Island, Portugal.
- Lownes, N. E., & Machemehl, R. B. (2006). *Vissim: A Multi-Parameter Sensitivity Analysis*. Paper presented at the Informs 2006 Winter Simulation Conference, California, USA.
- McKinnon, K. I. M. (1996). Convergence of the Nelder-Mead simplex method to a non-stationary point. *SIAM Journal on Optimization*, vol. 9 (1), pp. 148-158
- Miller, D. M. (2009). *Developing a Procedure to Identify Parameters for Calibration of a VISSIM Model*. Georgia Institute of Technology, Atlanta, USA.
- Morse, P., St-Aubin, P., Miranda-Moreno, L. F., & Saunier, N. (2016). *Transferability study of video tracking optimization for traffic data collection and analysis*, in Transportation Research Board Annual Meeting Compendium of Papers 2016.
- Nelder, J. A., & Mead, R. (1965). A Simplex Method for Function Minimization. *The Computer Journal*, vol. 7 (4), pp. 308-313
- ODOT. (2011). Protocol for VISSIM Simulation. Salem, USA: Oregon Department of Transportation.
- Oketch, T., & Carrick, M. (2005). *Calibration and Validation of a Micro-Simulation Model in Network Analysis*. Paper presented at the TRB Annual Meeting, Washington DC, USA.
- Otković, I. I., Tollazzi, T., & Šraml, M. (2013). Calibration of microsimulation traffic model using neural network approach. *Expert Systems with Applications*, vol. 40 (2013), pp. 5965–5974
- Ott, L., & Longnecker, M. (2010). *An introduction to Statistical Methods and Data Analysis, sixth edition*. Belmont, California: Brooks/Cole, Cengage Learning.
- Park, B. B., & Kwak, J. (2011). Calibration and validation of TRANSIMS microsimulator for an urban arterial network. *KSCE Journal of Civil Engineering*, vol. 15 (6), pp. 1091-1100
- Park, B. B., & Qi, H. M. (2005). Developement and Evaluation of a Procedure for the Calibration of Simulation Models. *Transportation Research Record*, vol. 1934, pp. 208-217

- Park, B. B., & Qi, H. M. (2006). Microscopic Simulation Model Calibration and Validation for Freeway Work Zone Network – A Case Study of VISSIM. *Proceedings of the IEEE ITSC 2006*, pp. 1471-1476
- Park, B. B., & Qi, H. M. (2007). Development and Evaluation of a Procedure for the Calibration of Simulation Models. *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1934, pp. 208-217
- Park, B. B., & Schneeberger, J. D. (2003). Microscopic Simulation Model Calibration and Validation - Case study of VISSIM Simulation Model for a Coordinated Actuated Signal System. *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1856, pp. 185-192
- PTV. (2013). PTV VISSIM 6, User Manual, version 13.11 (pp. 740). Karlsruhe, Germany.
- Saltelli, A., Tarantola, S., Campolongo, F., & Ratto, M. (2004). *Sensitivity Analysis in Practice - A guide to Assessing Scientific Models*. Chichester, England: John Wiley & Sons, Ltd.
- Saunier, N., & Sayed, T. (2006, 07-09 June 2006). *A feature-based tracking algorithm for vehicles in intersections*. Paper presented at the Computer and Robot Vision, 2006. The 3rd Canadian Conference on.
- Toledo, T., & Koutsopoulos, H. N. (2004). Statistical validation of traffic simulation models. *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1876 (1), pp. 142-150
- Toledo, T., Koutsopoulos, H. N., Davol, A., Ben-Akiva, M. E., Burghout, W., Andréasson, I., Johansson, T., & Lundi, C. (2003). Calibration and validation of microscopic traffic simulation tools: Stockholm case study. *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1831, pp. 67-75
- TSS. (2008). Aimsun 8 Users' Manual (pp. 372). Barcelone, Espagne: TSS-Transport Simulation Systems.
- Yu, L., Li, X., & Zhuo, W. (2004). A Genetic Algorithm-Based Approach to the Calibration of VISSIM Using GPS Data. *Applications of Advanced Technologies in Transtportation Engineering*, vol. 2004, pp. 406-411

Zhizhou, W., Jian, S., & Xiaoguang, Y. (2005). *Calibration of VISSIM For Shanghai Expressway Using Genetic Algorithm*. Paper presented at the 2005 Winter Simulation Conference, Orlando, Florida.

## **ANNEXE A – FONCTIONNEMENT DE CATS**

### **Traitement vidéo**

C'est la partie de CATS qui permet l'assignation des véhicules sur les différentes voies de circulation (voir section 3.2.4.3), permet de visualiser les trajectoires calculées par Traffic Intelligence (voir section 3.2.4.1) et de tracer à l'écran la position des différents points des alignements. Il est également possible de demander une analyse des trajectoires pour ressortir les véhicules avec une vitesse trop élevée ou étant détectés comme allant en sens inverse (vitesse négative) et de les exclure de l'analyse au besoin.

### **Interface de communication**

Cet élément de CATS permet la communication avec VISSIM. Ses fonctionnalités principales sont le démarrage et l'arrêt de VISSIM, ainsi que l'obtention et l'initialisation des valeurs des différents paramètres du logiciel. Un dictionnaire de la syntaxe d'appel des fonctionnalités de VISSIM est également inclus.

### **Détermination du nombre de répétitions à effectuer**

Tel que discuté à la section 0, cette portion de CATS met en œuvre la méthodologie tirée de (Hollander & Liu, 2008a) afin de déterminer le nombre minimal de répétitions à effectuer. Le résultat de cette étude donne un graphique tel que celui de la Figure 4-11 .La décision finale quant au nombre de répétitions étant sujette à l'analyse tirée de du graphique, elle est laissée entièrement à l'utilisateur.

### **Calibration**

Cette portion de CATS se divise en deux grandes sections : l'initialisation et la calibration. Les fonctions d'initialisation servent à mettre en place l'environnement de travail nécessaire pour que le logiciel d'optimisation, NOMAD (voir section 3.2.2), accomplisse son travail. Plus précisément, le fichier des paramètres de NOMAD est revu et corrigé afin d'initialiser la calibration selon les caractéristiques des paramètres à calibrer (type de donnés, bornes de recherche, etc.). Ces informations sont décidées par l'utilisateur dans le fichier de calibration de CATS. Ultimement, cette première section démarre NOMAD.

Les fonctions de calibration sont celles qui effectuent réellement le travail « d'adaptateur de la boîte noire » (Figure 3-1). L'information pertinente quant à la structure de l'étude de calibration telle que définie par les choix faits dans le fichier de configuration active ou désactive des pans de calcul et modifie le temps total de calcul. La fonction de calibration est écrite de sorte à être vue comme la boîte noire par NOMAD et fixe, via l'interface de communication, les paramètres de la simulation, charge les fichiers de sortie de VISSIM, et transforme les données chargées selon les procédés décrits aux sections 3.2.1.2, 3.3.1, 4.2 et 4.2.1. Plusieurs des calculs réutilisent la structure de données de Traffic Intelligence afin de disposer des observations terrain et des données simulées dans un même format et d'ainsi faciliter leur comparaison. Ces fonctions utilisent le module *subprocess* permettant de paralléliser les tâches et d'ainsi réduire le temps total nécessaire au calcul des points, soit un jeu de valeurs à appliquer à l'ensemble paramètres inclus dans la calibration, choisis par NOMAD dans le processus de calibration.

## **Validation**

Cette portion reprend tout ce qui se fait par la portion calibration, mais sans appeler NOMAD entre les deux sous-sections. Elle sert à valider les paramètres calibrés.

## **Analyse de sensibilité**

Cette portion de CATS sert à analyser les paramètres du logiciel de microsimulation. Pour l'instant, la seule méthode d'analyse de sensibilité disponible est le « un à la fois ». Dans sa forme actuelle, cette analyse permet d'inclure un grand nombre de paramètres et de les tester sur plusieurs points entre le minimum et le maximum défini par l'utilisateur alors que les autres paramètres restent fixes. Il est alors possible d'évaluer l'effet de chaque paramètre. Même si c'est la version la plus faible de l'analyse de sensibilité, elle était suffisante pour les besoins du projet. Cette portion pourra être bonifiée avec des méthodes plus poussées d'analyses de sensibilité.

## **Visualisation**

Plusieurs programmes indépendants permettent de visualiser certains aspects de la calibration. Ainsi, une de ces programmes permet de charger l'historique de la calibration et de tracer la « Visualisation de l'espace de recherche exploré par NOMAD », soit un graphique semblable à la Figure 5-1. Cette figure peut alors être utilisée pour prendre les décisions nécessaires pour initialiser une nouvelle calibration, tel que décrit à la section 3.2.5. Un autre programme permet de

tracer le graphique « Distributions cumulées des TIV [de suivi/de changement de voie] pour les vidéos utilisées lors des étapes de calibration et de validation », à ceux de la Figure 5-2 et de la Figure 5-3. Ce programme peut faire la visualisation pour tout point de la calibration, les plus intéressants étant bien évidemment le point de départ et l'optimum trouvé. Finalement, un dernier programme permet de tracer la « Visualisation de la progression des différentes composantes de la fonction objectif en fonction du nombre de jeux de paramètres testés par NOMAD », semblable à la Figure 5-4 et qui permet de déterminer le rôle des différentes composantes de la fonction objectif durant la calibration.

### **Librairies Python essentielles**

- |  |   |
|--|---|
| <i>Matplotlib</i>  | <a href="http://matplotlib.org/">http://matplotlib.org/</a>   |
| <ul style="list-style-type: none"> <li>• Traitement vidéo</li> <li>• Détermination du nombre de répétitions à effectuer</li> </ul> |   |
| <i>win32com.client</i>   | <a href="http://starship.python.net/~skippy/win32/Downloads.html">http://starship.python.net/~skippy/win32/Downloads.html</a> |
| <ul style="list-style-type: none"> <li>• Interface de communication</li> </ul>   |   |
| <i>Pandas</i>  | <a href="http://pandas.pydata.org/">http://pandas.pydata.org/</a>   |
| <ul style="list-style-type: none"> <li>• Calibration</li> <li>• Validation</li> <li>• Visualisation</li> </ul>                     |   |

### **Dépendances externes**

- |  |   |
|--|---|
| <i>Traffic Intelligence</i>  | <a href="https://bitbucket.org/Nicolas/trafficintelligence">https://bitbucket.org/Nicolas/trafficintelligence</a> |
| <ul style="list-style-type: none"> <li>• Analyse de sensibilité</li> <li>• Calibration</li> <li>• Détermination du nombre de répétition à effectuer</li> <li>• Traitement vidéo</li> <li>• Validation</li> </ul> |   |
| <i>Logiciel de microsimulation</i>   |   |
| <ul style="list-style-type: none"> <li>• Interface de communication</li> </ul>   |   |
| <i>Algorithme d'optimisation</i>   |   |
| <ul style="list-style-type: none"> <li>• Interface de communication</li> </ul>   |   |

## ANNEXE B – DESCRIPTION DES PARAMÈTRES DE VISSIM

### Paramètres du modèle de poursuite

Paramètre	Description
w99cc0	Distance moyenne souhaitée entre véhicules arrêtés. Il n'y a aucune variation
w99cc1	Intervalle véhicules en secondes qu'un conducteur souhaite conserver en fonction d'une certaine vitesse de circulation. Plus la valeur augmente, plus le conducteur est prudent. Pour une vitesse $v$ [m/s] prédéfinie, la distance de sécurité moyenne est calculée comme suit : $dx\_safe = CC0 + CC1 \cdot v$ <p>Dans le modèle de succession des véhicules, la distance de sécurité est définie comme la distance de succession minimale d'un conducteur par rapport au véhicule qui le précède. Cette valeur est déterminante pour la capacité d'une voirie dans un contexte de circulation dense.</p>
w99cc2	Limite les oscillations de la distance par rapport au véhicule précédent permise par un conducteur avant de s'en rapprocher sciemment. Pour une valeur de 10 m par exemple, il en résulte des distances de $dx\_safe$ à $dx\_safe + 10m$ pour le comportement de succession. La valeur par défaut de 4 m entraîne un comportement de conduite stable.
w99cc3	Définit le début de la décélération en secondes avant d'atteindre la distance de sécurité. Un conducteur perçoit un véhicule précédent plus lent à cet instant.
w99cc4 w99cc5	<b>CC4</b> définit la différence de vitesses négative au cours du processus de succession des véhicules <b>CC5</b> définit la différence de vitesses positive au cours du processus de succession des véhicules. Saisissez une valeur positive pour <b>CC5</b> qui correspond à la valeur négative de <b>CC4</b> . Des valeurs faibles entraînent une réaction plus sensible du conducteur à l'accélération ou à la décélération du véhicule précédent
w99cc6	Influence de l'interdistance sur l'oscillation de la vitesse au cours du processus de succession des véhicules <ul style="list-style-type: none"> <li>➤ Valeur 0 : l'oscillation est indépendante de l'interdistance</li> <li>➤ Valeurs élevées : l'oscillation de la vitesse est fonction croissante de l'interdistance des véhicules</li> </ul>
w99cc7	Accélération pendant l'oscillation
w99cc8	Accélération souhaitée départ arrêté, limitée par l'accélération maximale des courbes d'accélération.
w99cc9	Accélération souhaitée à 80 km/h, limitée par l'accélération maximale des courbes d'accélération.
obsrvdVeh	Nombre de véhicules qu'un conducteur peut anticiper pour réagir aux véhicules devant ou à côté de lui sur le même tronçon. Les véhicules tiennent compte de la distance de perception en plus du nombre de véhicules précédents saisi. Les véhicules perçoivent également les éléments de réseau suivants comme un véhicule précédent. Ces éléments de réseau sont alors interprétés comme un signal lumineux rouge. Si plusieurs de ces éléments de réseau sont situés sur une courte distance, augmentez le nombre de véhicules précédents. Ceci peut entraîner une légère augmentation du temps de calcul de la simulation. <ul style="list-style-type: none"> <li>➤ Zones de ralentissement</li> <li>➤ Signaux lumineux rouges</li> </ul>

## Paramètres du modèle de poursuite (suite)

<b>obsrvdVeh (suite)</b>	<ul style="list-style-type: none"> <li>➤ Perturbations liées au trafic transversal, lorsque l'interdistance minimale et l'intervalle véhicules minimal ne sont pas respectés.</li> </ul> <p>Les véhicules perçoivent les éléments de réseau suivants comme un véhicule précédent s'ils doivent s'y arrêter :</p> <ul style="list-style-type: none"> <li>➤ Panneaux Stop</li> <li>➤ Stations TC</li> <li>➤ Parkings</li> </ul> <ul style="list-style-type: none"> <li>➤ Comportement pour les zones de conflits : un véhicule tient compte de toutes les zones de conflits jusqu'au véhicule précédent indiqué par le nombre de véhicules précédents.</li> </ul>
------------------------------	---

*Texte tiré et adapté de (PTV, 2013) p.187,190-191*

## Paramètres du modèle de changement de voie

Paramètre	Description
<b>MaxDecelOwn</b> <b>MaxDecelTrail</b>	Saisir la décélération maximale pour le changement de voie en raison des itinéraires spécifiés pour le véhicule dépassant <b>MaxDecelOwn</b> et le véhicule suivant <b>MaxDecelTrail</b> Limite supérieure
<b>AccDecelOwn</b> <b>AccDecelTrail</b>	Limite inférieure (de la décélération)
<b>DecelRedDistOwn</b> <b>DecelRedDistTrail</b>	La différence de décélération est spécifiée en plus (en mètres par $-1 \text{ m/s}^2$ ). Elle diminue la <b>Décélération max.</b> de la valeur indiquée de manière linéaire jusqu'à la <b>Décélération acceptée</b> pour une distance croissante de la position d'arrêt d'urgence.
<b>DiffusTm</b>	Temps d'attente maximal d'un véhicule en situation d'arrêt d'urgence souhaitant procéder à un changement de voie. Après ce délai, le véhicule est exclu du réseau et une alerte correspondante est consignée dans le fichier *.err.
<b>MinHdwy</b>	Distance minimale entre deux véhicules après un changement de voie permettant ce changement de voie.
<b>SafDistFactLnChg</b>	Pris en considération pour chaque changement de voie. Il concerne les paramètres suivants : <ul style="list-style-type: none"> <li>➤ la distance de sécurité du véhicule suivant sur la nouvelle voie pour la décision si le changement de voie a lieu</li> <li>➤ la distance de sécurité du véhicule changeant de voie</li> <li>➤ la distance au véhicule changeant de voie précédent et moins rapide</li> </ul> Au cours du changement de voie, Vissim diminue la distance de sécurité en appliquant la valeur résultant de la multiplication suivante : <b>distance de sécurité initiale • coefficient pour distance de sécurité réduite</b> Avec la valeur par défaut 0,6, la distance de sécurité est diminuée de 40 %. Dès qu'un changement de voie est achevé, la distance de sécurité initiale est de nouveau prise en considération.

*Texte tiré et adapté de (PTV, 2013) p.194-195*