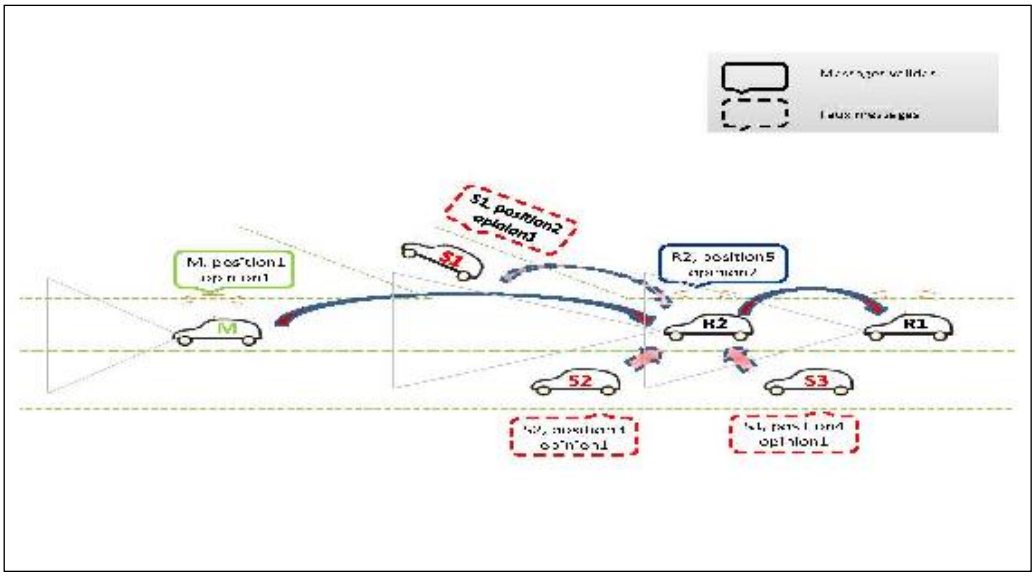
Fusion distribuée de données échangées dans un réseau de véhicules

Nicole El Zoghby

Par **Nicole EL ZOGHBY**

Fusion distribuée de données échangées dans un réseau de véhicules

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC



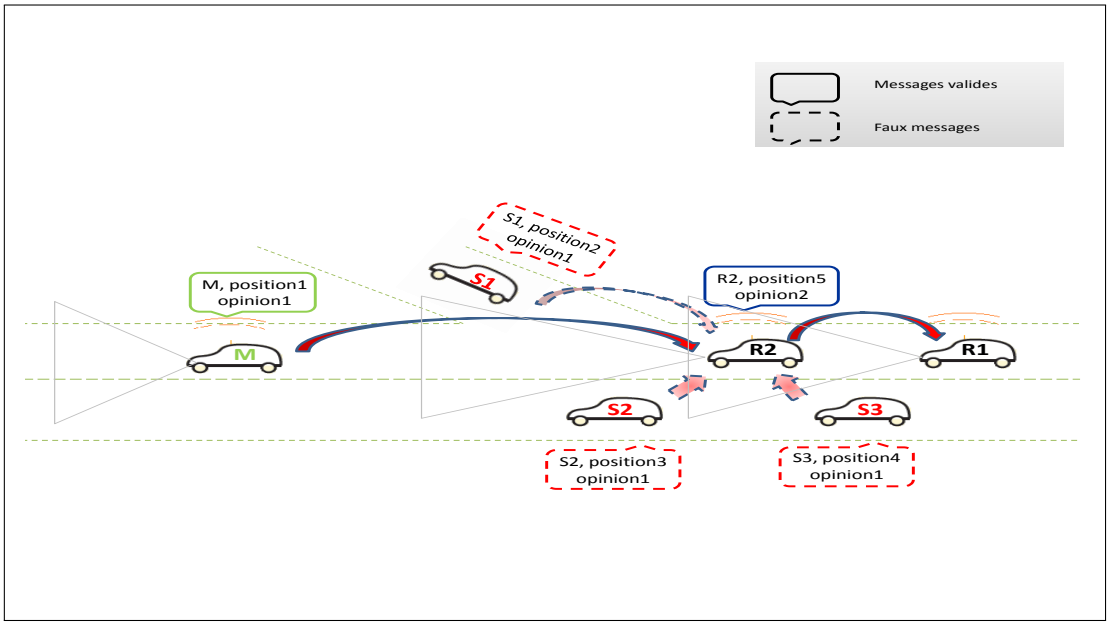
Soutenue le 19 février 2014
Spécialité : Technologies de l'Information et des Systèmes

D2133

par Nicole EL ZOGHBY

Fusion distribuée de données échangées dans un réseau de véhicules

Thèse présentée pour l'obtention du grade
de Docteur de l'UTC



Soutenue le : 19 février 2014
Spécialité : Technologie de l'Information et des Systèmes

Fusion distribuée de données échangées dans un réseau de véhicules

Nicole EL ZOGHBY

Thèse soutenue le 19 février devant le jury composé de :

Président :

Fawzi NASHASHIBI
Directeur de Recherche
INRIA Rocquencourt

Rapporteurs :

Michèle ROMBAUT
Professeur
Université Joseph Fourier

Eric LEFEVRE
Professeur
Université D'Artois

Examinateurs :

Bertrand DUCOURTHIAL
Professeur
Université de Technologie de Compiègne

Directeurs de Thèse :

Véronique CHERFAOUI
Maître de Conférences HDR
Univ. de Technologie de Compiègne

Thierry DENOEUX
Professeur
Univ. de Technologie de Compiègne

Université de Technologie de Compiègne

Laboratoire Heudiasyc UMR CNRS 7253

19 février 2014

A ma famille,

A Vous,

Bassam, Hélène et Joe,

pour votre support, affection, confiance et vos prières.

Remerciements

Il me sera difficile de remercier tout le monde car c'est grâce à l'aide de nombreuses personnes que j'ai pu mener cette thèse à son terme.

Je tiens à remercier l'ensemble du personnel du laboratoire Heudiasyc et plus particulièrement Ali Charara pour son acceuil au sein du laboratoire et ses conseils avisés qui m'ont permis de surmonter mes difficultés.

Je remercie aussi l'ensemble des membres du jury, Mme Michèle Rombaut et Mr. Eric Lefèvre pour leurs rapports incluant des remarques pertinentes vis à vis de mon travail, Mr. Fawzi Nashashibi pour son intérêt et sa présidence de mon jury et Mr. Bertrand Ducourthial pour sa collaboration qui a enrichi ce travail.

Je voudrais également exprimer ma profonde reconnaissance à Véronique Cherfaoui et Thierry Denoeux, mes directeurs de thèse, qui ont dirigé mon travail ; Leurs conseils et leurs commentaires précieux m'ont permis de progresser dans ma thèse.

Un grand merci à mes parents et mon frère J pour leur support et leur encouragement tout au long de cette thèse, ils ont cru en moi et finalement, maintenant j'y suis !

Il m'est impossible d'oublier de remercier ma petite famille libanaise : Zahra, Ali et Joseph, Reine, Oussama et Farah, Jessy, Bassam et Farah. Heureusement que vous êtiez là pour me changer les idées et pour m'encourager dans mes hauts et mes bas.

Ces remerciements seraient incomplets si je n'en adressais pas à mes collègues qui sont devenus mes amis. Je les remercie pour les moments agréables qu'on a partagés ensemble, pour les discussions scientifiques et non scientifiques. Vous êtes nombreux, je vais essayer de vous citer mais excusez moi si j'ai oublié quelqu'un. Merci Sawsan, Hoda, Vincent, Julien, Ji, Clément, Felipe, Nicolas, Bin, Gilles, Adam, Marek, Bihao, Zui, Yue, Kun... Je remercie mes amis au Liban, à Nancy et à Paris qui ont été toujours là pour moi.

Enfin, je remercie Sylvain pour son soutien quotidien indéfectible et son enthousiasme contagieux à l'égard de mes travaux comme de la vie en général.

14 février 2014
Nicole

Publications liées à la thèse

Articles publiés à une revue internationale

– T. Denoeux, N. El Zoghby, V. Cherfaoui, A. Jouglet. Optimal object association in the Dempster-Shafer framework. IEEE Transactions on Cybernetics. To appear.

Publications dans des conférences internationales :

– N. El Zoghby, V. Cherfaoui, B. Ducourthial and T. Denoeux. Distributed Data fusion for detecting Sybil attacks in VANETs. In T. Denoeux and M.-H. Masson (Eds), Belief functions : theory and applications. Proc. of the 2nd Int. Conf. on Belief Functions, Springer, AISC 164, Compiègne, France, 9-11 May 2012, pages 351-358.

– N. El Zoghby, V. Cherfaoui and T. Denoeux. Optimal object association from pairwise evidential mass functions. In Proceedings of the 16th Int. Conf. on Information Fusion (FUSION '13), Istanbul, Turkey, 9-12 July 2013, pages 774-780.

– N. El Zoghby, V. Cherfaoui and T. Denoeux. Evidential Distributed Dynamic Map for Cooperative Perception in VANets. In the proceedings of IEEE intelligent Vehicles Symposium 2014 (IV'14), Dearborn, Michigan, USA, June 8 - 11, 2014.

Publications dans des conférences nationales :

– N. El Zoghby, V. Cherfaoui, B. Ducourthial et T. Denoeux. Fusion distribuée évidentielle pour la détection d'attaques sybil dans un réseau de véhicules. In Rencontres Francophones sur la Logique Floue et ses Applications (LFA 2012), pages 63-70, Compiègne, France, November 2012, Cépaduès-Editions.

Chapitre de livre :

– N. El Zoghby, V. Loscri, E. Natalizio, V. Cherfaoui. Robot cooperation and swarm intelligence. "Wireless Sensor and Robot Networks - From Topology Control

to Communication Aspects", N. Mitton and D. Simplot-Ryl (Eds.), published by World Scientific. Edited by : Nathalie Mitton (Inria Lille – Nord Europe, France), David Simplot-Ryl (Inria Lille – Nord Europe, France), February 2014.

Table des matières

Table des figures

Liste des tableaux

Introduction

0.1. Contexte de la thèse

Les véhicules seront bientôt capables de communiquer entre eux et de s'organiser sous forme de réseau. Ils pourront ainsi échanger des informations importantes pour la sécurité et le confort du conducteur. De nombreuses applications d'aide à la conduite et gestion de trafic peuvent ainsi être envisagées. De récents projets internationaux (CVIS, Safespot, Coopers, SEVECOM, Grand Cooperative Driving Challenge, DRIVE C2x, SIM-TD) et le consortium européen "Car 2 Car Communication" sont des illustrations de l'investissement des constructeurs et des pouvoirs publics dans ce domaine.

Ainsi les VANets : "Vehicular Ad Hoc Networks" sont des réseaux ad-hoc sans fil où les véhicules sont considérés comme les nœuds d'un réseau ad-hoc sans fil capables de s'organiser sans infrastructure définie préalablement. Les relais sont tout simplement les véhicules qui acheminent l'information de l'un à l'autre sans avoir à passer par un maillage d'émetteurs et récepteurs externes pour couvrir le territoire. Un nouveau standard de communication entre véhicules et équipements de bord de route est actuellement déployé : il s'agit du IEEE 802.11p. Il permet l'échange d'information entre les véhicules et entre les véhicules à grande vitesse et l'infrastructure.

Le projet PRE-DRIVE C2X [dri] propose de classer en trois catégories les fonctions envisagées grâce à la communication véhicule-véhicule (C2C) ou la communication véhicule -infrastructure (C2I) :

– les fonctions relatives à la sécurité : systèmes d'aide à la conduite pour accroitre la sécurité routière en réduisant le nombre d'accidents ou l'impact des accidents non évitables. On citera, par exemple, la prévention de collision, le freinage d'urgence, l'alerte d'accidents, etc.

- les fonctions relatives à la gestion du trafic : contrôle de la congestion du trafic afin de réduire les temps de transport et la consommation de carburant, la gestion de flottes de véhicules, la conduite en convoi ou la gestion de carrefours intelligents sont des exemples de fonctions rendues possibles grâce aux communications C2X.
- les fonctions relatives au divertissement et activités commerciales : accès à des services d'informations permettant aux passagers une activité de divertissement ou de travail.

Les principaux problèmes à gérer dans les réseaux mobiles ad-hoc sont d'une part liés à la nature du réseau : perte de données, perte de chemins, sécurité limitée, erreurs de transmission, nœuds cachés, etc, et d'autre part liés au traitement des données échangées : définition du contenu et du format des informations à transmettre, confiance qu'on peut accorder aux messages, méthode de traitement pour synthétiser l'information. C'est sur ces derniers points que nous avons focalisé notre étude.

0.2. Sujet de thèse

Echanger des données dans un VANet d'une manière sécurisée implique l'introduction de la notion de confiance. Chaque noeud doit avoir confiance dans les autres noeuds ou dans les données reçues avant d'utiliser les informations échangées pour ses propres applications. Chaque véhicule peut être autonome pour estimer la confiance mais la coopération entre les véhicules permet d'améliorer et de rendre plus robuste cette estimation. On retrouve ces principes dans les systèmes multi-agents. Cette thèse porte sur l'étude des techniques de fusion de données réparties et incertaines au sein d'un réseau mobile pour gérer la confiance. L'objectif de cette étude est, d'une part, de définir un opérateur de combinaison réparti et robuste et, d'autre part, d'identifier le contenu des informations échangées dans le réseau en fonction des applications envisagées. Il s'agit de fusionner les informations transmises par d'autres véhicules pour arriver à un état de connaissance global et avoir un réseau robuste vis à vis des attaques et des pannes. Le problème théorique de la thèse relève donc de la fusion distribuée. Ce problème s'avère difficile dans le cadre des VANets à cause de la nature du réseau, de la mobilité des nœuds et de la connectivité entre les nœuds.

Nous proposons dans cette thèse un algorithme de fusion distribuée dans le cadre des fonctions de croyance. L'algorithme est appliqué par chaque noeud à la réception des messages et permet de fusionner les données afin de gérer la confiance. Nous nous sommes intéressés à deux types de sous- problèmes : la gestion de la confiance dans les noeuds du réseau et la gestion de la confiance dans les données échangées dans le réseau.

Dans le premier cas, nous nous intéressons à la détection de faux nœuds créés par un nœud malveillant dans le réseau. Dans un réseau de véhicules par exemple, la création de faux véhicules par un véhicule malveillant peut avoir des conséquences importantes. L'opérateur de combinaison réparti et robuste pourrait être une alternative aux solutions de sécurité classiques (PKI, authentification...). Ces solutions peuvent difficilement être déployées dans des réseaux n'ayant pas d'accès aisé à l'infrastructure de bord de route. Le but d'une telle application est de quantifier la confiance dans les noeuds de réseau. L'algorithme de fusion distribuée permet de détecter les faux noeuds créés par le noeud malveillant.

La deuxième application est la fusion de cartes locales dynamiques distribuées. L'objectif est de fournir aux véhicules communicants une "perception augmentée" de leur environnement afin d'envisager des aides à la conduite plus adaptées. La gestion des incertitudes est un point clé de ce type d'application. Il s'agit de quantifier la confiance dans l'existence des objets détectés dans une scène. Nous traitons dans cette application les données échangées dans le réseau.

Comme dans tout problème de fusion, distribuée ou non, se pose le problème de l'association. C'est une étape préalable à la fusion pour mettre en correspondance les différentes données. L'algorithme d'association développé durant cette thèse consiste à trouver une relation entre deux ensembles d'objets. Il intervient dans l'application de fusion de cartes dynamiques distribuées. Ce problème est aussi formalisé dans le cadre des fonctions de croyance.

0.3. Plan du manuscrit

Dans ce mémoire, nous allons évoquer les différents outils utilisés pour pouvoir définir un opérateur de combinaison réparti et robuste. Le chapitre 1 présente les bases de la théorie des fonctions de croyance qui est la théorie principale utilisée dans cette thèse pour modéliser et gérer la confiance. Nous abordons la fusion dis-

tribuée, ses particularités dans le réseau dynamique. Le chapitre 2 présente notre contribution dans le domaine de la fusion distribuée. Après un état de l'art sur la gestion de la confiance dans les réseaux fixes et mobiles, nous avons proposé un algorithme permettant de combiner les messages reçus sur la base des techniques de fusion de données. Nous évoquons ensuite les différentes applications que nous avons développées durant cette thèse : l'attaque sybil (chapitre 3) et la carte dynamique distribuée (chapitre 4). La fusion distribuée appliquée à l'attaque sybil permet de détecter les faux noeuds créés par un noeud malveillant. Cette application a été validée par simulation. La carte dynamique distribuée est une adaptation de l'algorithme de fusion distribuée afin d'augmenter le champ de perception des véhicules en coopérant. La simulation de différents scénarios de véhicules équipés détectant des objets dans une scène a permis la validation de cette application. L'approche d'association d'objets, une étape préalable à la fusion, est aussi présentée dans le chapitre 5. Des exemples sur des données simulées et réelles ont permis de valider cette approche. Nous terminerons ce manuscrit par une conclusion générale et nous présentons les perspectives de ces travaux.

CHAPITRE 1

La théorie des fonctions de croyance

1.1. Introduction

Les fonctions de croyance constituent un des principaux cadres pour raisonner avec des informations incertaines et imprécises. Ce cadre, introduit par Dempster (1968) et Shafer (1976), généralise les mesures de probabilité et les mesures de possibilité.

Les travaux de Dempster [Dem67] sur les bornes inférieures et supérieures d'une famille de distributions de probabilités ont permis à Shafer [Sha76] de formaliser les bases de la théorie des fonctions de croyance. Shafer a montré l'intérêt des fonctions de croyance pour la modélisation des connaissances incertaines. Ph. Smets a contribué, via le modèle de croyance transférables ([SK94], [Sme00]), à populariser la théorie développée par Dempster et Shafer.

La théorie des fonctions de croyance a deux composantes principales : la représentation de l'information sous formes de fonctions de masse, de crédibilité et de plausibilité et une règle de combinaison pour combiner les éléments d'évidence indépendants.

Ce chapitre expose les bases de cette théorie qui sera utilisée dans le cadre de cette thèse.

1.2. Le Modèle des Croyances Transférables

Une fonction de croyance s'interprète dans le modèle des croyances transférables, développé par Smets, comme une opinion pondérée, un degré de croyance d'un capteur ou d'un agent en charge du raisonnement. Dans le cadre de MCT, deux niveaux sont distingués dans la modélisation du raisonnement de l'agent rationnel en charge de la prise de décision :

– le niveau crédal, où sont représentées et manipulées les informations disponibles.
– le niveau décisionnel, siège de la construction de la décision de l'agent.

Cette décomposition en deux niveaux différencie le MCT des autres théories et méthodes, en particulier la théorie des probabilités au sein de laquelle la représentation des connaissances, la combinaison et la décision sont effectuées sans distinguer les niveaux d'abstraction.

Dans un cadre général, on cherche à identifier une hypothèse parmi un ensemble d'hypothèses possibles. Cet ensemble d'hypothèses décrit toutes les solutions du problème à traiter. La théorie de l'évidence permet d'évaluer la véracité de propositions. Pour se faire, elle se fonde sur des degrés de croyances pour représenter l'incertitude sur les différentes propositions.

1.2.1. Représentation de l'information :

Soit $\Omega = \{w_1, ..., w_k, ..., w_K\}$ un ensemble fini, généralement appelé cadre de discernement, contenant l'ensemble des solutions possibles. Une fonction de croyance peut être définie par une fonction de masse, notée m définie de 2^Ω dans $[0, 1]$ qui vérifie :

$$\sum_{A \subseteq \Omega} m(A) = 1. \tag{1.1}$$

Chaque sous-ensemble $A \subseteq \Omega$ tel que $m(A) > 0$ est appelé élément focal de m. Ainsi, la masse $m(A)$ représente le degré de croyance attribué à la proposition A et qui n'a pas pu, compte tenu de l'état de la connaissance, être affecté à un sous-ensemble plus spécifique que A.

Il existe quelques cas particuliers concernant les fonctions de masse. Nous proposons quelques définitions, à savoir une fonction de masse est :

– **normale** si et seulement si $m(\emptyset) = 0$, $\emptyset$ n'est pas un élément focal de Ω,

- **catégorique** si et seulement si $m(A) = 1$ et on notera m_A la fonction de masse catégorique d'ensemble focal A,
- **vide** si et seulement si $m(\Omega) = 1$, c'est la masse catégorique d'ensemble focal Ω, notée m_Ω,
- **bayésienne** si et seulement si $m(A) = 0$, $\forall A/|A| > 1$, c'est à dire si tous les éléments focaux sont des singletons de Ω,
- **dogmatique** si et seulement si $m(\Omega) = 0$,
- **consonante** si et seulement si tous ses éléments focaux sont emboîtés,
- **simple** si et seulement si elle a seulement deux éléments focaux A et Ω, tel que $A \subset \Omega$ et $A \neq \emptyset$,
- **spécialisée** si et seulement si elle a seulement trois éléments focaux A, $\bar{A}$ et Ω, tel que $A \cup \bar{A} = \Omega$ et A, $\bar{A} \neq \emptyset$.

Étant donnée une fonction de masse m, on peut définir la crédibilité Bel sur l'ensemble $2^\Omega \to [0, 1]$ comme :

$$Bel(A) = \sum_{\emptyset \neq B \subseteq A} m(B),\ \forall A \subseteq \Omega. \tag{1.2}$$

La quantité $Bel(A)$ mesure à quel point les informations données par une source soutiennent la proposition A. La *crédibilité* donne ainsi une tendance minimale.

La fonction de *plausibilité* associée à m est la fonction de 2^Ω dans $[0, 1]$ définie par :

$$Pl(A) = \sum_{B \cap A \neq \emptyset} m(B),\ \forall A \subseteq \Omega. \tag{1.3}$$

La quantité $Pl(A)$ représente le degré maximal de croyance susceptible d'être alloué à A après intégration de nouvelles informations. La plausibilité donne ainsi une tendance maximale. La fonction $pl : \Omega \to [0, 1]$ tel que $pl(w) = Pl(\{w\})$ pour tout $w \in \Omega$ est appelée une fonction de contour associée à m.

Par ailleurs, soit la fonction $q : 2^\Omega \to [0, 1]$ définie par

$$q(A) = \sum_{B \supseteq A} m(B),\ \forall A \subseteq \Omega. \tag{1.4}$$

Cette fonction est appelée fonction de *communalité*. Elle associe à toute partie A de Ω la masse restant attachée à A après conditionnement par ce même ensemble.

Toutes ces fonctions représentent de différentes manières une même information.

1.3. Combinaison de l'information

1.3.1. Opérateur de Dempster

Soient deux masses m_1 et m_2 définies sur Ω. Ces deux fonctions peuvent être agrégées par un opérateur de combinaison conjonctif noté $\textcircled{\cap}$. Le résultat de cette opération conduit à une fonction de masse unique de la manière suivante :

$$m_{\textcircled{\cap}}(A) = (m_1 \textcircled{\cap} m_2)(A) = \sum_{B \cap C = A} m_1(B) m_2(C) \quad \forall A \subseteq \Omega. \tag{1.5}$$

Cette règle conjonctive est parfois nommée règle de combinaison de Dempster non normalisée. Si nécessaire, l'hypothèse de normalisation $m_{\textcircled{\cap}}(\emptyset) = 0$ peut être retrouvée en divisant chaque masse par un coefficient adéquat. L'opérateur résultant, qui est connu sous le nom de *règle de Dempster* et noté $m_{\oplus}$, est défini par :

$$(m_1 \oplus m_2)(A) = \frac{1}{1-\kappa} \sum_{B \cap C = A} m_1(B) m_2(C) \tag{1.6}$$

pour tout $A \subseteq \Omega$, $A \neq \emptyset$ et $(m_1 \oplus m_2)(\emptyset) = 0$, où

$$\kappa = \sum_{B \cap C = \emptyset} m_1(B) m_2(C) \tag{1.7}$$

est le *degré de conflit* entre m_1 and m_2. Si $\kappa = 1$, il y a une contradiction logique entre les deux éléments d'évidence et ceux-ci ne peuvent pas être combinés. La règle de Dempster a un élément neutre, la fonction de masse *vide* m_Ω définie par $m(\Omega) = 1$.

La règle de Dempster a une propriété intéressante relative aux fonctions de contour. Si pl_1 et pl_2 sont les fonctions de contour de deux fonctions de masse $m_1$et m_2, en utilisant le même symbole $\oplus$, la fonction de contour de $m_1 \oplus m_2$ est alors :

$$(pl_1 \oplus pl_2)(w) = \frac{pl_1(w).pl_2(w)}{1-\kappa}, \tag{1.8}$$

pour tout $w \in \Omega$.

A noter que la communalité de la masse résultante d'une combinaison conjonctive

est obtenue par multiplication des fonctions de communalité :

$$q_{\textcircled{\cap}}(A) = q_1(A).q_2(A) \quad \forall A \subseteq \Omega. \tag{1.9}$$

La règle conjonctive et celle de Dempster sont commutatives et associatives, ce qui facilite la combinaison des sources. On peut donc combiner les sources dans n'importe quel ordre et de façon récursive. Ces propriétés sont très importantes dans le cadre de la fusion de données multicapteurs car on ne contrôle pas toujours l'ordre d'arrivée des données.

1.3.2. Opérateur prudent de Denoeux

La règle de Dempster suppose l'indépendance des sources d'informations. Autrement dit, la même information ne doit pas être comptée deux fois. D'où la nécessité de disposer de règles de combinaison tolérant la dépendance et la redondance des informations combinées. Pour qu'une règle de combinaison $*$ puisse combiner des sources non distinctes, elle doit impérativement être idempotente $m * m = m$. La règle conjonctive prudente de Denoeux ([Den08], [Den06]) possède justement cette caractéristique. Elle repose sur le principe d'engagement minimal : lorsque plusieurs fonctions de croyance sont compatibles avec un ensemble de contraintes, la moins informative doit être choisie. Ce principe suppose l'existence d'une relation d'ordre entre les fonctions de croyances. Parmi ces relations d'ordre on peut citer $\sqsubseteq_w$, qui se base sur les poids w issus de la décomposition canonique décrite dans la section 1.4 ci-après.

Si m_1 et m_2 sont deux fonctions de masse non dogmatiques, on peut dire que m_1 est plus riche que m_2 au sens de $\sqsubseteq_w$, si $w_1(A) \leq w_2(A)$, pour tout $A \subset \Omega$. On notera alors $m_1 \sqsubseteq_w m_2$. Parmi toutes les fonctions de masses plus riches que deux fonctions de masses m_1 et m_2, le plus grand élément par la relation $\sqsubseteq_w$ existe et est unique Il est défini par :

$$w_{1\textcircled{\cap}2} = w_1 \wedge w_2 \quad \forall A \subset \Omega. \tag{1.10}$$

avec $\wedge = minimum$. L'opérateur prudent est défini de la façon suivante :

$$m_{1\textcircled{\cap}2} = m_1 \textcircled{\cap} m_2 = \bigoplus_{A \subset \Omega} A^{w_1(A) \wedge w_2(A)} \tag{1.11}$$

Cet opérateur est commutatif, associatif et idempotent.

1.4. Décomposition canonique

On appelle fonction de masse simple toute fonction $m : 2^\Omega \to \mathbb{R}$ qui vérifie :

$$\begin{array}{rcl} m(A) & = & 1 - w_A, \\ m(\Omega) & = & w_A. \end{array} \tag{1.12}$$

avec $A \subset \Omega$ et $w_A \in [0, 1]$. On note A^{w_A} cette fonction de masse. On a la propriété suivante :

$$A_1^{w_1} \textcircled{\cap} A_2^{w_2} = A^{w_1 w_2}. \tag{1.13}$$

Une fonction de masse est dite séparable si elle admet une décomposition de la forme :

$$m = A_1^{w_1} \textcircled{\cap} ... \textcircled{\cap} A_n^{w_n} = \textcircled{\cap}_{A \subset \Omega} A^{w_A} \qquad w_i \in [0, 1], i = 1, ..., n. \tag{1.14}$$

Si une fonction de masse m est non dogmatique ($m(\Omega) > 0$) et si les A_i sont tous distincts, la décomposition est alors unique (décomposition canonique). Il est possible donc d'écrire :

$$m = \textcircled{\cap}_{A \subset \Omega} A^{w_A} \quad avec\ w(A)\ \in\ [0, 1]\ pour\ tout\ A \subset \Omega \tag{1.15}$$

Toute fonction de masse séparable non dogmatique peut donc être représentée de façon unique par une fonction $w : 2^\Omega \to [0, 1]$. La fonction $w : 2^\Omega \setminus \{\Omega\} \to [0, +\infty[$ est une nouvelle représentation de m (fonction de pondération conjonctive). Les poids $w(A)$ pour tout $A \in 2^\Omega \setminus \{\Omega\}$ sont obtenus par :

$$w(A) = \prod_{B \supseteq A} q(B)^{(-1)^{|B|-|A|+1}} = \begin{cases} \dfrac{\prod_{B \supseteq A, |B| \notin 2\mathbb{N}} q(B)}{\prod_{B \supseteq A, |B| \in 2\mathbb{N}} q(B)} & \text{si} \quad |A| \in 2\mathbb{N} \\ \dfrac{\prod_{B \supseteq A, |B| \in 2\mathbb{N}} q(B)}{\prod_{B \supseteq A, |B| \notin 2\mathbb{N}} q(B)} & \text{sinon.} \end{cases} \tag{1.16}$$

1.5. Théorème de Bayes Généralisé

Le théorème de Bayes généralisé [Sme93a] permet de trouver la classe à laquelle appartient un objet o avec un attribut connu en se basant sur un ensemble d'apprentissage.

Considérons deux variables $X \in \Omega_X$ et $\theta \in \Theta = \{\theta_1, ..., \theta_k\}$. X est observable et peut être un résultat de mesure. θ est non observable, c'est une classe ou un paramètre inconnu à estimer. En observant $X = x$, il faut calculer la fonction de croyance sur Θ. On sait calculer $pl_k(x), \forall x, k$, la plausibilité que x appartient à la classe θ_k mais on n'a pas de connaissance à priori sur θ : $m^\Theta(\Theta) = 1$. La solution à un tel problème est une conséquence du principe d'engagement minimal :

$$m^\Theta[x] = \bigcirc\!\!\!\!\bigcap_{k=1}^{k} \overline{\{\theta_k\}}^{pl_k(x)}, \tag{1.17}$$

où $\bigcirc\!\!\!\!\bigcap$ représente la règle conjonctive et la notation $\{w\}^c$ représente la fonction de masse simple affectant la masse c à Θ et $1 - c$ à $\{w\}$.

1.6. Affaiblissement

En fusion d'informations, une source peut fournir un ensemble de valeurs qui ne sont pas totalement fiables. Il est donc important de prendre en compte la fiabilité de la source. Soit α le coefficient d'affaiblissement associé à la source S en fonction de la confiance qui lui est accordée :

- Si $\alpha = 0$, la source est totalement fiable : ${}^0m^\Omega = m_S^\Omega$.
- Si $\alpha = 1$, la source n'est pas du tout fiable : ${}^1m^\Omega = m_\Omega^\Omega$.

L'affaiblissement est appliqué sur la masse elle-même et produit une nouvelle masse de la façon suivante :

$${}^\alpha m^\Omega \quad = \quad (1 - \alpha).m_S^\Omega + \alpha.m_\Omega^\Omega. \tag{1.18}$$

Cette opération permet de pondérer l'importance de la fonction de masse lors de la fusion avec d'autres sources, sans l'éliminer complètement.

1.7. Probabilité pignistique

Comme nous avons déjà mentionné dans la section 1.2, le modèle des croyances transférable distingue deux niveaux : crédal et décisionnel. Pour passer d'un niveau à l'autre, il faut transformer les fonctions de masse en probabilité. Cette transformation est appelée la transformation pignistique et donne ce qu'on appelle la probabilité pignistique $BetP$ sous la forme suivante :

$$BetP(B) = \sum_{\emptyset \neq A \subseteq \Omega} \frac{|A \cap B|}{|A|} \frac{m(A)}{(1 - m(\emptyset))} \quad \forall B \subseteq \Omega. \tag{1.19}$$

La transformation pignistique peut être utile dans le cas où on veut comparer différentes mesures incertaines. La probabilité pignistique est utilisée dans la phase de décision pour choisir à chaque étape l'hypothèse la plus probable comme solution du problème.

1.8. Grossissement et raffinement

Soient deux cadres de discernement Ω et Θ. Le raffinement est l'opération qui à une hypothèse du cadre de discernement Θ, associe l'ensemble des hypothèses de Ω qui sont compatibles avec elle. Le grossissement constitue l'opération inverse.

Soit ρ la fonction qui associe l'ensemble des hypothèses. Elle est appelée raffinement de Θ si et seulement si :

- l'ensemble $\{\rho(\{\theta\}), \theta \in \Theta\} \subseteq 2^{\Omega}$ est une partition de Ω, et
- pour tout $A \subseteq \Theta$:

$$\rho(A) = \bigcup_{\theta \in A} \rho(\{\theta\}). \tag{1.20}$$

La figure 1.1 montre un exemple : Θ est un grossissement de Ω et Ω est un raffinement de Θ.

1.9. Conclusion

Les fonctions de croyance sont l'outil principal utilisé dans cette thèse pour élaborer un algorithme de fusion distribuée. Ce chapitre en a présenté un résumé pour montrer l'intérêt et la richesse de cette théorie. Le fait de transformer les données

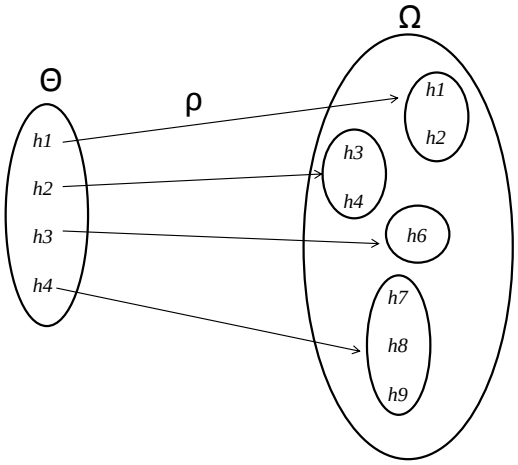


FIGURE *1.1.:* *Fonction de raffinement ρ*

réelles en fonctions de masses pour ensuite les combiner permet de gérer les imprécisions et les incertitudes.

Nous allons utiliser les fonctions de croyance pour la fusion, l'association et la mise à jour des données. Dans le chapitre suivant, nous allons montrer comment utiliser les fonctions de croyance pour fusionner les informations de manière distribuée afin de gérer la confiance dans les réseaux.

CHAPITRE 2

Gestion de confiance par la fusion distribuée

Nous nous intéressons à l'échange des informations dans un réseau de véhicules pour des applications de sécurité ou de gestion de trafic. Comme nous l'avons souligné précédemment, ceci nécessite de gérer la confiance dans les nœuds et dans les messages. La question qui se pose est comment gérer cette confiance d'une façon distribuée ? Afin de répondre à cette question, nous commençons par faire un état de l'art sur la gestion de confiance dans les réseaux en général. Nous présenterons ensuite la gestion de confiance par notre approche de fusion distribuée, utilisant les fonctions de croyance pour la gestion des incertitudes et des imprécisions.

2.1. Gestion de la confiance dans les réseaux

La gestion de la confiance s'applique aux réseaux fixes et mobiles et permet d'être plus robuste aux pannes et aux attaques. Certaines méthodes cherchent à estimer la confiance dans les nœuds du réseau, d'autres agrègent les données afin de profiter de la redondance des informations sans prendre en compte la nature de la source.

2.1.1. Quels sont les outils pour agréger les données ?

Plusieurs méthodologies ont été proposées pour agréger les données échangées dans un réseau. Parmi celles-ci, on peut citer les méthodes les plus courantes : le

vote majoritaire, la somme pondérée, la règle de Bayes et l'opérateur de Dempster. Nous rappelons ici rapidement le principe de ces techniques que nous illustrons ensuite avec un exemple extrait de [CV05].

Le vote majoritaire

Dans un système basé sur le vote, chaque nœud exprime son vote par envoi de message et le nœud récepteur effectue le traitement en fonction des avis recueillis. La valeur obtenue par la majorité sera le résultat de la combinaison.

La somme pondérée

La somme pondérée est une alternative au vote majoritaire. Chaque noeud peut donner un nombre entre 0 et 1 représentant son avis. Le résultat de la fusion sera la moyenne de ces nombres.

La règle de Bayes

La règle de Bayes se base sur la théorie de probabilités pour exprimer l'incertitude des données. Elle est utilisée pour mettre à jour les probabilités ou un paramètre quelconque, à partir des observations et des lois de probabilité de ces observations :

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}, \tag{2.1}$$

où $P(H|E)$ est appelée la probabilité a posteriori de H sachant E, c'est la mesure de la croyance en une hypothèse ou proposition H mise à jour sous condition E ; $P(H)$ est la probabilité a priori de H, qui reflète la croyance en H en absence de l'évidence E.

L'approche bayésienne nécessite des connaissances a priori, qui peuvent être difficile de déterminer dans la pratique. Le point faible d'une telle approche est la difficulté de modéliser l'absence d'informations.

L'opérateur de Dempster

Le manque de connaissance sur un évènement n'est pas nécessairement considéré comme un rejet de cet évènement. De plus, en présence de deux évènements,

l'incertitude sur l'un d'eux ne peut être considérée comme accréditant l'autre. La différence majeure avec la règle de Bayes est que l'opérateur de Dempster est plus adapté pour les cas des informations incertaines et imprécises.

Dans le cadre de Dempster-Shafer, la probabilité est remplacée par un intervalle d'incertitude borné par la croyance (belief) et la plausibilité. La croyance est la borne inférieure de cet intervalle et représente le degré de croyance en une observation. La plausibilité est la borne supérieure de l'intervalle et représente le degré maximal de croyance susceptible d'être alloué à une observation après intégration de nouvelles informations. Cet opérateur a été décrit dans la section 1.3.1 du chapitre 1.

Exemple

Pour guider le lecteur, nous allons donner un exemple qui met en valeur la différence entre ces différentes méthodes. Cet exemple est extrait de l'article [CV05]. Les auteurs considèrent une configuration avec quatre noeuds A, B, C et S. Les noeuds A, B et C communiquent entre eux et évaluent le noeud S. Ils peuvent ne pas être honnêtes. La figure 2.1 présente la configuration des quatre noeuds.

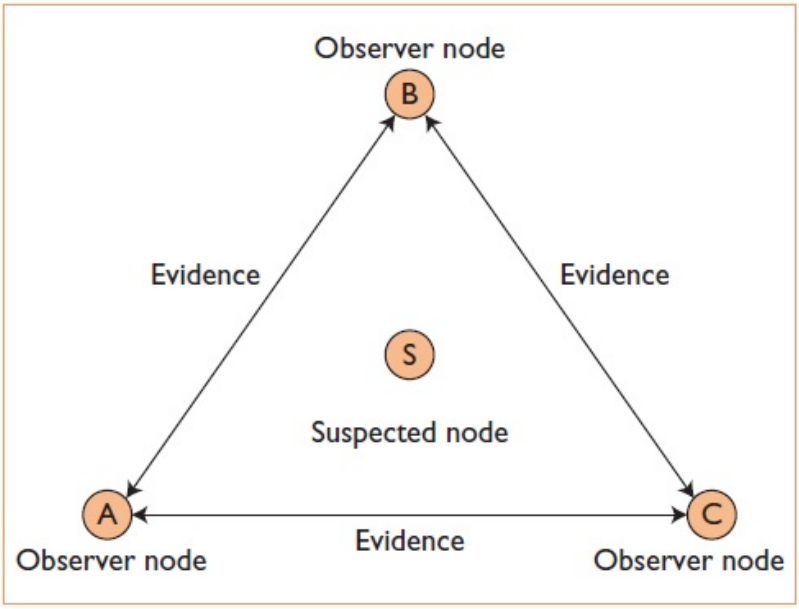


FIGURE 2.1.: *Exemple de scenario extrait de l'article [CV05].*

Les auteurs ont exploré trois cas pour montrer la différence entre les différentes

méthodes. Ils n'ont pas utilisé la règle de Bayes puisqu'ils n'ont pas de connaissance a priori généralement difficile à déterminer.

Cas 1 : Les auteurs supposent que les noeuds A, B et C sont honnêtes avec des probabilités 0.9, 0.8 et 0.2. A et B disent que S est honnête et C dit qu'il n'est pas honnête. Le tableau 2.1 montre la représentation des avis des noeuds A, B et C par la règle de Dempster, la somme pondérée et le vote majoritaire ainsi que le résultat sur la confiance en S.

TABLE 2.1.: *Cas 1 : Avis des noeuds A, B et C par la règle de Dempster, la somme pondérée et le vote majoritaire*

	Masse	Poids	Vote
A	$m_A(S) = 0.9$ $m_A(\overline{S}) = 0$ $m_A(\Omega) = 0.1$	0.9	1
B	$m_B(S) = 0.8$ $m_B(\overline{S}) = 0$ $m_B(\Omega) = 0.2$	0.8	1
C	$m_C(S) = 0$ $m_C(\overline{S}) = 0.2$ $m_C(\Omega) = 0.8$	0	0
Résultat	$m(S) = 0.975$	0.56	1

En utilisant l'opérateur de Dempster, la confiance du noeud S est de 0.975. Dans ce cas, le vote majoritaire va conclure à l'honnêteté du noeud S tandis que la somme pondérée donne un poids de 0.56 pour l'honnêteté de S.

Cas 2 : Les noeuds A, B et C sont honnêtes avec des probabilités 0.9, 0.2 et 0.2. A a confiance en S tandis B et C n'ont pas confiance. Le tableau 2.2 montre le résultat de la combinaison des avis des noeuds A, B et C par la règle de Dempster, la somme pondérée et le vote majoritaire ainsi que le résultat sur la confiance en S. En utilisant l'opérateur de Dempster, la confiance du noeud S est de 0.852. Dans ce cas, le vote majoritaire et la somme pondérée donnent un résultat différent.

Dans ces deux cas, les auteurs montrent que si les observations sont exactes, l'opérateur de Dempster écarte l'opinion des noeuds non honnêtes.

Cas 3 : Les noeuds A, B et C sont honnêtes avec une probabilité 0.8 mais leurs avis ne sont pas en accord. A et B ont confiance en S et C n'a pas confiance. Le

TABLE 2.2.: *Cas 2 : Avis des noeuds A, B et C par la règle de Dempster, la somme pondérée et le vote majoritaire*

	Masse	Poids	Vote
A	$m_A(S) = 0.9$ $m_A(\overline{S}) = 0$ $m_A(\Omega) = 0.1$	0.9	1
B	$m_B(S) = 0$ $m_B(\overline{S}) = 0.2$ $m_B(\Omega) = 0.8$	0	0
C	$m_C(S) = 0$ $m_C(\overline{S}) = 0.2$ $m_C(\Omega) = 0.8$	0	0
Résultat	$m(S) = 0.852$	0.3	0

tableau 2.3 montre la représentation des avis des noeuds A, B et C par la règle de Dempster, la somme pondérée et le vote majoritaire ainsi que le résultat sur la confiance en S.

TABLE 2.3.: *Cas 3 : Avis des noeuds A, B et C par la règle de Dempster, la somme pondérée et le vote majoritaire*

	Masse	Poids	Vote
A	$m_A(S) = 0.8$ $m_A(\overline{S}) = 0$ $m_A(\Omega) = 0.2$	0.8	1
B	$m_B(S) = 0.8$ $m_B(\overline{S}) = 0$ $m_B(\Omega) = 0.2$	0.8	1
C	$m_C(S) = 0$ $m_C(\overline{S}) = 0.8$ $m_C(\Omega) = 0.2$	0	0
Résultat	$m(S) = 0.828$	0.8	1

La combinaison par la règle de Dempster donne un résultat égal à 0.828. Le jugement dans ce cas suit la majorité.

2.1.2. Traitement des données sans prise en compte de la confiance dans la source

On considère ici les données échangées elles-mêmes, sans prendre en considération les sources d'informations. Différentes approches ont été développées dans ce but pour traiter différents types de problèmes.

Comme cela a été montré dans [CV05], dont l'exemple précédent a été extrait, le traitement de la confiance dans les données peut être basé sur la combinaison des informations échangées avec l'une ou l'autre méthode. Les auteurs n'ont pas traité le problème des connaissances a priori ni le cas de l'existence d'un noeud malveillant.

Cherfaoui et al. [CDC08] ont traité le problème de la gestion de la diffusion d'informations dans les VANets. Les auteurs ont développé un algorithme pour la combinaison des informations dans les systèmes distribués. Leur but est de traiter les évènements pour augmenter la visibilité du conducteur ; pour cela ils combinent le contenu des messages en prenant en compte les effets de l'espace et les retards. Le contenu du message est représenté par une masse de croyance. La méthode décrite considère un seul type d'évènement. Chaque véhicule, récepteur du message, décide s'il attribue la confiance en un évènement ou non. Dans cet article, les auteurs utilisent la règle prudente dans le cas des messages dépendants et la règle de Dempster dans le cas des messages indépendants. La thèse s'est basée sur ces travaux pour développer des algorithmes abordant des cas plus généraux.

Golle et al. [GGS04] ont considéré la validité des données. Ils utilisent la technique se basant sur les données des capteurs pour détecter ce qu'on appelle l'attaque "sybil" [1]. Chaque nœud possède un modèle du VANet composé d'un ensemble d'évènements localisés et de noeuds. Il peut améliorer ce modèle en ajoutant ses propres observations. Si toutes les données reçues par un nœud sont en accord avec le modèle du VANet, alors les données sont valides. Les auteurs considèrent que l'attaque d'un nœud malveillant est beaucoup plus probable que la collusion entre les nœuds. Leur technique pour détecter l'attaque se base sur la vérification de la position et la parcimonie contradictoire qui consiste à trouver la meilleure explication pour les données corrompues. Les auteurs utilisent des capteurs de position comme la caméra pour estimer la position relative et des capteurs pour le calcul du temps d'arrivée du signal et l'angle d'arrivée. Chaque nœud connaît la source physique du

1. L'attaque sybil est une des applications de cette thèse et sera traitée dans le chapitre 3.

message. Ils considèrent que la connectivité entre les nœuds est toujours possible et les nœuds génèrent des nouvelles clés, ce qui n'est pas favorable dans le cas de l'attaque "sybil". Les auteurs testent leur approche sur deux exemples. Dans le premier cas, les nœuds peuvent connaître la position exacte du nœud avec lequel ils peuvent communiquer. Dans le second exemple, les nœuds connaissent seulement la distance de leur voisin.

Selon Raya et al. [RPGH08], il est plus utile d'étudier la confiance dans les données que dans les nœuds eux-mêmes. Les nœuds peuvent avoir des niveaux de confiance préréglés (par exemple la voiture de police) mais différents évènements signalés par un même véhicule peuvent avoir différents niveaux de confiance. Les auteurs étudient la confiance des données délivrées par les nœuds. Ils calculent la confiance de chaque donnée, et cette croyance est ensuite combinée pour prendre une décision finale. Leur système est composé des éléments suivants :

- un ensemble d'évènements Ω,
- un ensemble de nœuds $V = v_k$. Chaque nœud appartient à un type $\theta_n \in \Theta$,
- une valeur de confiance par défaut est définie pour chaque nœud,
- chaque nœud est responsable d'une tâche.

Les auteurs définissent différentes variables :

- f est la confiance attribuée à un évènement. Cette variable dépend du nœud et de sa tâche. $f \in [0, 1]$.
- Les facteurs de confiance dynamiques : $s(v_k)$ et μ_l. $s(v_k)$ est une variable binaire qui représente le statut de sécurité : lorsqu'elle est égale à un, le nœud est considéré comme légitime. Sinon, le nœud est considéré comme non légitime. μ_l indique si les attributs d'un nœud ont changé (par exemple le lieu d'un nœud). μ_l est une valeur réelle comprise entre 0 et 1.

Pour calculer la confiance dans un rapport généré par un nœud v_k et fournissant des éléments d'évidence sur un évènement, une valeur F est calculée en fonction des trois variables précédentes : $F(f, \mu_l, s(v_k))$ et $F \in [0, 1]$. Cette valeur est appelée "weight" ou "trust level". Chaque évènement rapporté et son poids respectif calculé, peuvent servir comme entrée pour le modèle de décision logique. La sortie du module de décision logique est le niveau de confiance sur ces données. La décision logique est implémentée en utilisant plusieurs techniques, comme le vote, la règle de Bayes et l'opérateur de Dempster. Les auteurs comparent ces différentes techniques. Les deux techniques les plus utilisées sont l'approche bayésienne et la théorie de Dempster-Shafer. L'approche bayésienne nécessite des connaissances

préalables et la théorie de Dempster Shafer permet de travailler avec les données incertaines. Donc selon le scénario spécifique, l'une ou l'autre des techniques peut être utilisée.

2.1.3. Prise en compte de la confiance dans les nœuds

Ces méthodes prennent en considération les sources d'informations. Elles traitent la confiance dans les nœuds d'un réseau par deux approches principales : les mécanismes de réputation et l'évaluation de la confiance ("trust").

Mécanismes de réputation

Les mécanismes de réputation se basent sur l'évaluation des nœuds pour gérer la confiance dans un système qui varie selon le type d'application envisagée. Plusieurs types de problèmes doivent être considérés comme le changement d'identité des utilisateurs ou la création de faux nœuds. Ce type de mécanisme est largement étudié dans la communauté de l'internet.

Dans [ZM00], les auteurs s'intéressent aux mécanismes de réputation pour les communautés "en ligne". Ils essaient de résoudre le problème de la gestion de la confiance dans les systèmes multi-agents, où les agents sont des êtres humains à objectifs financiers. Deux mécanismes de réputation ont été développés : Sporas et Histos. Sporas est dédié aux systèmes qui n'ont pas une forte connectivité. Chaque nouvel utilisateur débute avec une valeur minimale de réputation et cette valeur est mise à jour après chaque évaluation selon l'activité de l'utilisateur dans le système. Dans Sporas, deux utilisateurs s'évaluent l'un l'autre une seule fois. La valeur de la réputation est calculée sous forme d'une somme pondérée en fonction des nombres de réputation et de la valeur de la réputation donnée par l'utilisateur. En ce qui concerne le système de réputation Histos, il nécessite une forte connexion entre les utilisateurs. Les auteurs modélisent leur système comme un graphe dont les nœuds représentent les utilisateurs et les arêtes pondérées la dernière évaluation faite entre deux utilisateurs. Chaque utilisateur cherche le lien direct avec un autre utilisateur. Une fois ce lien trouvé, la réputation personnelle de chaque nœud est faite. Leur système cherche donc les chemins directs entre deux utilisateurs, de longueur inférieure à un certain seuil. Pour évaluer la valeur de la réputation personnelle d'un utilisateur, on peut se baser sur l'avis des autres utilisateurs du nœud

de l'itération précédente (il y a une relation de récursivité). Les auteurs ont validé leur approche par simulation sur des marchés de services.

Un autre mécanisme de réputation est développé dans [LI04]. Les auteurs ont décrit un mécanisme de réputation pour les MANets (Mobile Ad hoc Networks ou réseau Ad hoc mobiles) avec deux dimensions importantes : le temps et le contexte. Leur approche consiste à considérer la réputation, à l'évaluer et à la propager dans le système. Ils utilisent la somme pondérée comme outil pour agréger les données.

Leur modèle de réputation est basé sur différents éléments :

– $SRep$: Service Reputation ;
– $RRep$: Recommendation Reputation.

Chaque agent développe le $SRep$ d'un autre agent selon son expérience $SExp$ et les recommandations Rec. La réputation dépend du temps et change selon les activités. La notion de temps et de contexte (importance et utilité du service, catégorie du service, etc.) est prise en compte dans le calcul de la réputation. La réputation est affaiblie en fonction du temps et du contexte.

Chaque nœud a un gestionnaire d'expérience, un gestionnaire de recommandation et un gestionnaire de réputation. Le gestionnaire d'expérience enregistre les expériences en notant le temps, le contexte et une valeur agrégée de l'expérience ($SExp$). Le gestionnaire de recommandations enregistre les recommandations des autres noeuds, échange les informations de réputation avec les autres agents et gère la table $RRep$. Le gestionnaire de réputation calcule $SRep$ en prenant en considération les entrées du gestionnaire de l'expérience et de la recommandation. Des interactions se déroulent dans le système. Après chaque interaction, chaque agent donne un score de satisfaction pour l'interaction ; les agents mettent à jour la valeur de l'expérience $SExp$ de la réputation $SRep$ et la recommandation.

Pour la propagation de la réputation, le gestionnaire de recommandation contacte les différents agents pour l'échange des informations de réputation. Pour pouvoir échanger, ces agents doivent avoir un $RRep$ supérieur à un seuil prédéfini.

Les auteurs expliquent leur approche en indiquant que leur mécanisme permet de se défendre contre trois types d'attaques : l'inactivité, c'est le cas des agents qui refusent de partager la réputation avec les autres ; la diffamation, qui consiste à choisir une victime et lui donner une mauvaise réputation ; la collusion où les agents propagent des bonnes réputations pour se favoriser les uns des autres. Leur mécanisme renforce le partage actif et sécurisé des informations de réputation. Le

point faible de ce système de réputation est l'absence de défense contre le changement des identités, généralement effectué par un nœud malveillant qui crée de fausses identités afin d'acquérir une bonne réputation. Cette méthode a été validée par simulation sur 100 agents parmi lesquels 30 sont dignes de confiance dans le service et les recommandations, 30 sont dignes de confiance dans les recommandations et non dans le service, les autres 40 agents étant indignes de confiance dans les deux cas.

Pour résumer, les mécanismes de réputation se basent sur la notion d'évaluation d'un agent par différents évaluateurs du réseau après des expériences d'interaction directe avec lui et des recommandations des voisins de l'évaluateur qui ont interagi avec ce même agent. On retrouvera ce principe, qui est la base des applications réparties, dans notre algorithme de fusion distribuée.

Évaluation de la confiance ("trust")

La confiance ("trust") est une relation entre les différentes entités d'un système qui participent à différents protocoles. L'évaluation du "trust" se fait souvent suite à des interactions entre les différents nœuds du réseau.

Theodorkopoulos et al. [TB04] ont établi les relations de confiance indirecte entre les nœuds d'un graphe sans interaction directe. Ils ont présenté un schéma pour l'évaluation des preuves de confiance ("trust evidence") dans les réseaux Ad hoc.

Ils considèrent les hypothèses suivantes :
- absence d'infrastructure préalable,
- absence d'infrastructure des clés publiques,
- les preuves sont incertaines et incomplètes,
- En présence des attaquants, une partie du réseau peut être inaccessible.

Dans leur système, ils considèrent deux problèmes :
- déterminer l'opinion que peut avoir un nœud sur un autre en utilisant l'opinion d'un nœud intermédiaire,
- trouver le chemin de confiance (trusted path) entre deux nœuds, celui qui a la plus grande valeur de "trust".

Leur approche se base sur une structure algébrique appelé semi-anneau basée sur un triplet $(S, \bigoplus, \bigotimes)$ où S est l'ensemble des nœuds, $\bigoplus$ et $\bigotimes$ sont des opérateurs binaires. $\bigoplus$ est commutatif, associatif et a $\textcircled{0}$ comme élément neutre. $\bigotimes$

est associatif, a ① comme élément neutre et ⓪ comme élément absorbant. Ces opérateurs sont choisis sous différentes formes selon l'application visée. Le semi-anneau doit être ordonné $(S, \bigoplus, \bigotimes, \leq)$ et idempotent $\forall a \in S, a \bigoplus a = a$. Les auteurs définissent deux types de semi-anneau : chemin, distance. Ils décrivent leur algorithme "generic-single-source-shortest-distance (G-S)" qui est une extension de l'algorithme de Dijkstra. Les auteurs valident leur approche par simulation d'un réseau fixe et comparent trois topologies "Random, Grid, Smallworld". Ce qui caractérise ce travail est la possibilité de détection des attaques dans le système et la modélisation de l'incertitude.

Un autre modèle de confiance est proposé dans [WS09]. Ce modèle est basé sur la théorie de Dezert-Samarandache (DSmT), une extension de la théorie des fonctions de croyance. Leur approche est proposée pour les communautés ouvertes où chaque membre du réseau peut rejoindre et quitter le réseau à n'importe quel moment. Ceci cause le problème de manque d'informations sur les différents membres d'où la nécessité d'évaluer la confiance. Les auteurs représentent celle-ci en prenant en considération les données incertaines et contradictoires. Les croyances sont présentes dans $(T, \neg T, T \cap \neg T, \Omega)$ où T signifie "trust" , $\neg T$ "distrust", $T \cap \neg T$ la contradiction et Ω supporte l'ignorance. Les auteurs comparent leur travail avec celui de Yu et Singh [BY02] qui se base sur la théorie de Dempster Shafer. La démarche des auteurs se base sur quatre étapes :

- l'évaluation des noeuds est faite suite à un certain nombre d'interactions entre deux noeuds. Selon le service demandé, un indicateur de QoS (Quality of Service) est attribué à chaque service ;
- la confiance est calculée par transitivité et ils utilisent pour le calcul un affaiblissement $m(T) + \alpha m(T \cap \neg T)$. Le choix de cet affaiblissement n'est pas détaillé ;
- les confiances calculées par transitivité sont combinées par la règle de Dempster ;
- la décision est prise selon l'interaction entre deux nœuds (confiance locale), la confiance globale provenant des voisins et l'insuffisance des informations entre deux agents.

Leur démarche ne traite pas le cas où la même information est combinée plusieurs fois et l'affectation d'une masse à la proposition $(T \cap \neg T)$ n'est pas justifiée.

Il est nécessaire d'étudier la nature de la source avant de fusionner les données

qu'elle envoie. La gestion de la confiance des sources de l'information s'avère importante étant donné que cette dernière peut être un attaquant qui essaie de modifier les fonctionnalités du réseau.

2.2. Fusion distribuée dans un réseau de véhicules

Nous avons vu que, pour la gestion de la confiance dans un réseau, les méthodes varient selon qu'elles traitent les sources ou les informations. Nous proposons de mettre à profit les méthodes de fusion de données dans des applications réparties. En effet les techniques de fusion de données proposent des outils permettant de modéliser et de prendre en compte les incertitudes sur les données et sur les sources d'information. Comme nous l'avons montré dans le chapitre 1, le cadre des fonctions de croyances est particulièrement adapté à cette problématique grâce à sa capacité à représenter les incertitudes et à la diversité des opérateurs de combinaison. La difficulté réside dans sa mise en oeuvre dans un système distribué et de configuration dynamique.

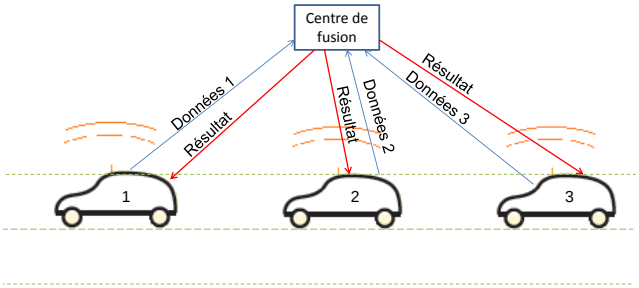
Nous allons tout d'abord définir la fusion distribuée et ses particularités dans les réseaux dynamiques, puis nous montrons comment, en utilisant les fonctions de croyances, nous pouvons définir un opérateur de fusion réparti et robuste.

2.2.1. Définition

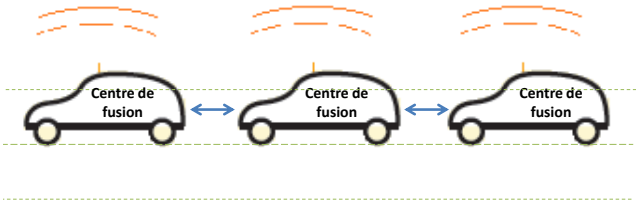
La fusion de données est un ensemble de méthodes permettant de combiner des données provenant de sources différentes afin de prendre une décision et de réduire les incertitudes sur le résultat final. Dans le domaine de véhicules intelligents, les sources d'informations sont les capteurs et la fusion est souvent faite dans un centre de fusion qui collecte ces informations ([Mit07], [LHL08]). La fusion peut être centralisée ou décentralisée (distribuée).

Le centre de fusion centralisé est commun à tous les véhicules (figure 2.2a). Il collecte les données de tous les véhicules du réseau, les fusionne et envoie le résultat.

La **fusion centralisée** souffre des retards de communication puisque toutes les unités du réseau communiquent avec le même centre. De plus, l'augmentation du nombre de véhicules augmente la charge de calcul. Cette architecture est difficile à



(a) Fusion centralisée



(b) Fusion décentralisée

FIGURE *2.2.: Fusion centralisée et décentralisée.*

déployer dans les réseaux de véhicules car elle nécessite des infrastructures spécifiques.

La **fusion décentralisée** ou **distribuée** se fait souvent dans le véhicule lui-même, les informations fusionnées étant envoyées à l'ensemble du réseau (figure 2.2b). Dans ce cas, chaque véhicule est muni d'un algorithme de fusion qui traite localement les informations. Cette architecture convient à la dynamique du réseau de véhicules mais dans certains cas elle peut conduire à traiter des données déjà fusionnées puisque la même information peut être reçue plusieurs fois.

L'étape de fusion de données centralisée ou décentralisée est souvent précédée d'une étape d'**association de données**. Cette étape permet de mettre en corres-

pondance des données dans le but de les combiner. Elle peut intervenir à plusieurs niveaux selon l'application envisagée. Elle peut être appliquée sur :

- les observations à différents instants,
- les informations détectées par différents capteurs sur un même véhicule,
- des informations détectées par différents véhicules, dans le cas où chaque véhicule est muni d'un ou plusieurs capteurs et envoie ce qu'il détecte aux autres.

Le problème d'association sera traité dans le chapitre 5. Dans ce qui suit, nous allons évoquer la particularité de la fusion dans les réseaux dynamiques et présenter ensuite notre contribution vis-à-vis de ces particularités.

2.2.2. Particularités de la fusion dans un réseau dynamique

La fusion distribuée dans les VANets rencontre plusieurs problèmes liés à la nature du réseau d'une part et à l'application de la fusion en elle-même d'autre part. Les VANets sont constitués de véhicules capables de s'échanger des informations dans le but d'améliorer la sécurité routière ou de permettre l'accès à internet pour les passagers. L'échange des informations se fait via une connexion wifi, ce qui engendre des problèmes de transmissions sans fil, parmi lesquels on peut citer la perte de la connexion, la perte des données ou la perte de chemins.

Par rapport aux réseaux ad hoc classiques, les VANets se différencient par une forte mobilité des nœuds rendant la topologie du réseau fortement dynamique, ce qui risque d'engendrer une déconnexion du graphe. A ces problèmes s'ajoute celui de la dissémination par cycle, les mêmes informations étant reçues plusieurs fois par le même nœud, ce qu'on appelle "Data incest" [MKC03]. Plusieurs solutions ont été présentées pour régler les problèmes de dissémination par cycle et gérer les informations dépendantes. Les principales méthodes trouvées dans la littérature sont l'utilisation de matrices d'informations [ASKB12], l'intersection de covariance ([CAM02], [JU01]), la règle prudente ([Den08], [Den06]) et les méthodes ensemblistes ([JKDW01], [JL08]).

Du point de vue de la fusion de données, il s'agit aussi de traiter les problèmes de la synchronisation de temps et de la gestion de l'espace. Les nœuds étant mobiles, chaque nœud doit décider, à chaque réception du message, de la véracité du message en se basant sur l'emplacement de l'émetteur, le temps d'envoi du message et le temps de traitement, s'il s'agit de fusionner plusieurs messages. Ces aspects seront abordés principalement dans les applications (chapitre 3 et 4).

2.3. Principe de l'algorithme de fusion distribuée

Nous considérons un système de fusion distribuée composé d'un ensemble de nœuds communicants entre eux, chaque nœud ayant une connaissance partielle sur l'ensemble des nœuds du réseau. La communication des informations locales se fait par l'échange des messages entre les nœuds. Chaque nœud fusionne les informations reçues, ce qui rend le système de fusion décentralisé. Ce système doit gérer les incertitudes et les imprécisions ainsi que les données déjà fusionnées à cause des cycles possibles dans le réseau.

Pour développer un algorithme de fusion distribuée permettant de gérer les confiances dans un réseau de véhicules en tenant compte des données redondantes, nous nous sommes appuyés sur les travaux du domaine des algorithmes distribués. Nous nous sommes basés sur les travaux de Ducourthial [Duc07] sur les r-opérateurs qui assurent l'auto-stabilisation d'un système distribué. Il s'agit de trouver un opérateur de fusion auto-stabilisant, réparti et robuste.

Nous considérons un réseau constitué de nœuds échangeant des messages, représenté par un graphe direct $G = (V, A_r)$ où V est l'ensemble des nœuds $V = \{v_1, v_2, ..., v_n\}$ et A_r est l'ensemble d'arrêtes. L'ensemble des voisins du nœud v_i à portée d'antenne est noté $\Gamma(v_i) = \{v_j \in V, \{v_i, v_j\} \in A_r\}$. Pour simplifier, nous supposons que chaque nœud connaît le nombre de noeuds $n = |V|$. Il faut noter que l'algorithme permet à chaque noeud d'acquérir les informations sur les noeuds en dehors du voisinage correspondant à la portée de l'antenne ($\Gamma(v)$). L'algorithme traite les données d'une manière asynchrone. Nous proposons une méthodologie de fusion de données dans un réseau pour combiner les données échangées dans un réseau ad hoc mobile.

Dans le cadre de la fusion distribuée, nous considérons que chaque noeud a un avis sur les noeuds du réseau ou sur les données échangées. Cette confiance est représentée par une fonction de masse notée m, qui peut être vide en cas d'incertitude (m_Ω).

2.3.1. Connaissances gérées par les noeuds

Chaque noeud dispose de trois types de connaissance dont deux sont mémorisées. Il a un moyen d'acquérir une connaissance directe, notée $C_{directe_i}(t)$, indépendante des informations qu'il reçoit par message. Cette information peut provenir d'un dis-

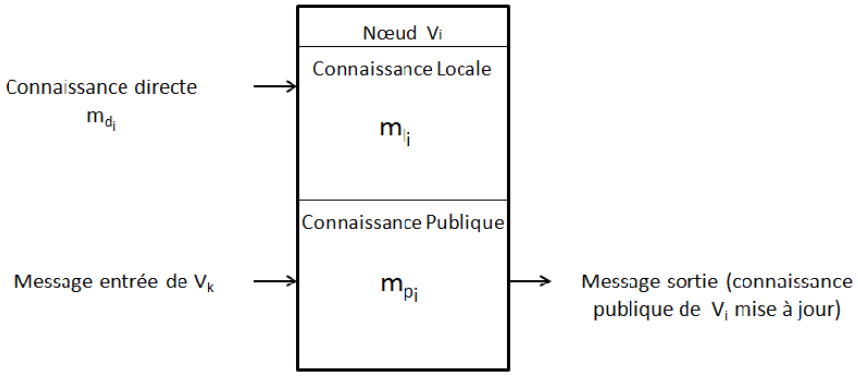


FIGURE 2.3.: *Schéma représentant les différentes connaissances d'un noeud V_i*

positif externe (capteur embarqué, antenne, autre algorithme, etc.). Le modèle de cette connaissance est défini selon l'application envisagée. Il s'agit de transformer une mesure réelle en fonction de masse $m_{d_i}^{(t)}$. Les croyances directes sont conservées dans une mémoire locale appelée connaissance *locale*. Cette partie *locale* est combinée avec ce qui provient des autres nœuds pour mettre à jour une connaissance *publique* ou *distribuée* qui sera rediffusée à travers le réseau. Cette distinction permet de séparer ce qui provient des mesures directes de ce qui est calculé par le réseau. Ce principe correspond à un système distribué [Duc07]. Ainsi la mémoire interne d'un noeud v_i à l'instant t contient deux vecteurs de masse représentant chacun une connaissance :

$$\begin{aligned} C_{locale_i}(t) &= m_{l_i}^{(t)}, \\ C_{publique_i}(t) &= m_{p_i}^{(t)}. \end{aligned} \tag{2.2}$$

Les différentes connaissances d'un noeud V_i sont représentées dans la figure 2.3

2.3.2. Algorithme de fusion

Cas d'un seul élément d'observation

L'algorithme 1 décrit les différentes étapes de la fusion distribuée à la réception d'un message de l'émetteur v_k contenant un seul élément d'observation représenté par la fonction de masse $m_{p_k}^{(t)}$. Il est représenté par la figure 2.4. Pour faciliter la compréhension de notre approche, nous allons utiliser un exemple tout au long de

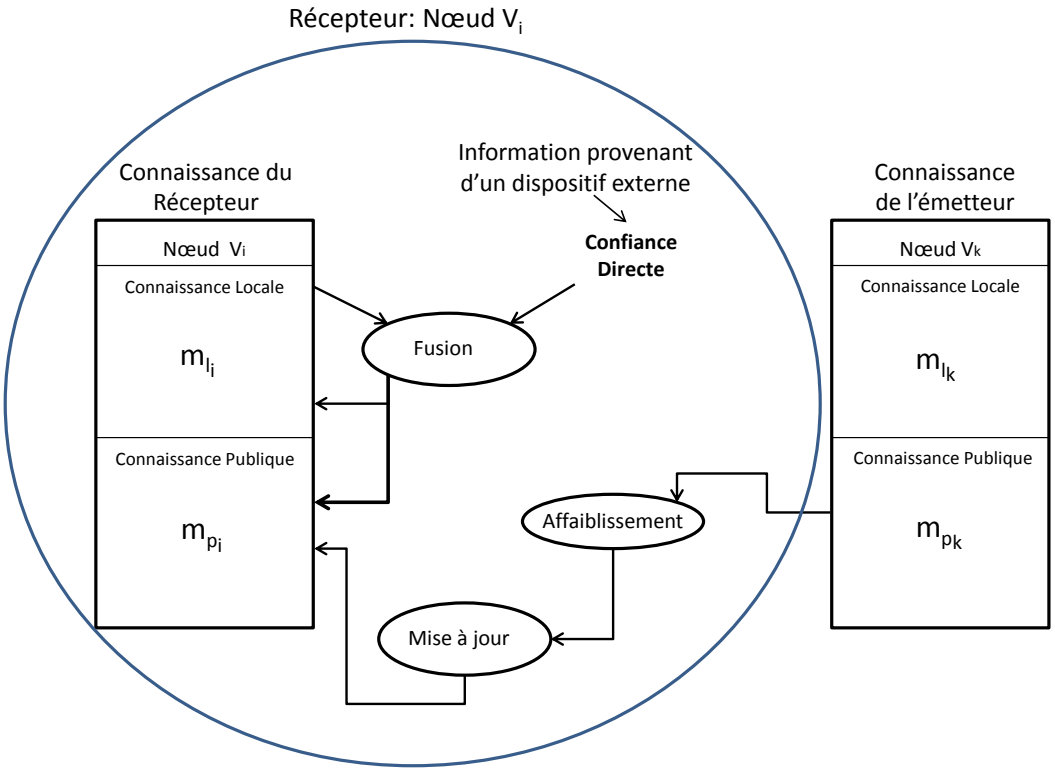


FIGURE 2.4.: *Schéma de description de l'algorithme de Fusion distribuée*

cette partie.

Exemple : *Imaginons un ensemble de quatre noeuds v_A, v_B, v_C et v_D qui s'échangent des messages concernant un événement E qui correspond au fait qu'il ait du brouillard ou non. On suppose que le cadre de discernement est $\Omega = \{E, \overline{E}\}$, où $m(\emptyset)$=0, $m(E)$ représente la confiance dans l'évènement E et $m(\overline{E})$ la confiance que l'événement est faux, $m(\Omega)$ représente l'incertitude dans l'évènement E.Chaque noeud est muni d'un dispositif externe lui permettant d'avoir une masse sur cet évènement. Le dispositif du noeud v_C donne des informations erronées. v_A ne détecte pas l'événement E. La figure 2.5 représente un exemple de configuration de ces quatre noeuds.*

Chaque nœud v_i calcule la confiance directe $m_{d_i}^{(t)}$ sur l'événement (ligne 3 de l'algorithme). Cette confiance est indépendante des messages précédents et n'est pas le résultat d'autres combinaisons. Le modèle de cette information varie selon l'application envisagée et sera détaillé dans les chapitres 3 et 4.

Algorithme 1 : Fusion distribuée à la réception d'un message sur le noeud i : cas d'une masse sur un seul élément

Données : Message de l'émetteur v_k contenant $m_{p_k}^{(t)}$;
Résultats : C_{locale_i}=$m_{l_i}^{(t)}$ et $C_{publique_i}$=$m_{p_i}^{(t)}$;
$m_{d_i}^{(t)} \leftarrow ConfianceDirecte()$;
Mise à jour de la connaissance locale
$m_{l_i}^{(t)} \leftarrow m_{l_i}^{(t-1)} \oplus m_{d_i}^{(t)}$;
Mise à jour de la connaissance publique
Avec la connaissance locale
$m_{p_i}^{(t)} \leftarrow m_{p_i}^{(t-1)} \textcircled{\&} m_{l_i}^{(t)}$;
$\alpha \leftarrow CalculAffaiblissement()$;
${}^{\alpha}m_{p_k}^{(t)} \leftarrow (1-\alpha).m_{p_k}^{(t)} + \alpha.m_{\Omega}^{(t)}$;
Avec le message
$m_{p_i}^{(t)} \leftarrow m_{p_i}^{(t)} \textcircled{\&}^{\alpha} m_{p_k}^{(t)}$;
Envoi message contenant $m_{p_i}^{(t)}$;

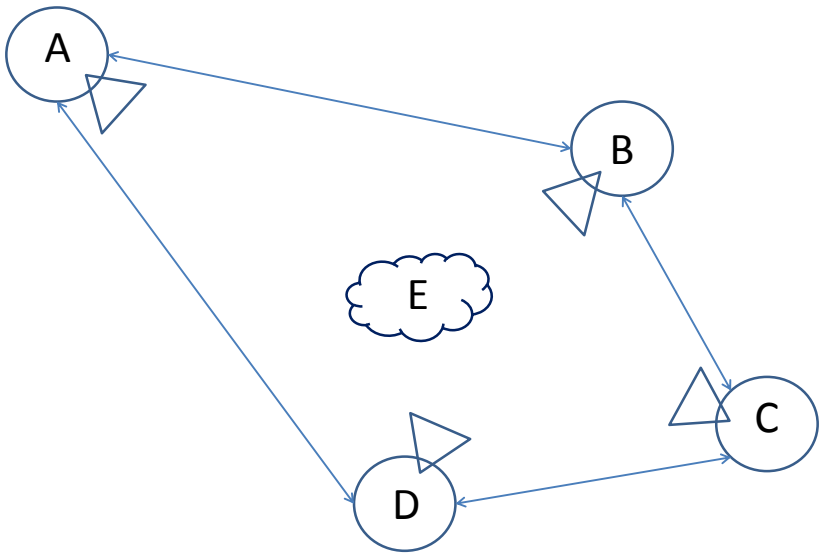


Figure 2.5.: *Exemple de configuration : A, B, C et D observent un évènement E. Les messages sont échangés entre les noeuds sauf entre A et C et entre D et B.*

Exemple : *Soit la confiance directe $m_{d_i}^{(t)}$ des quatre noeuds de l'exemple à un instant initial noté t :*

$$\begin{cases} m_{d_A}^{(t)}(E) = 0 \\ m_{d_A}^{(t)}(\overline{E}) = 0 \\ m_{d_A}^{(t)}(\Omega) = 1 \end{cases} \quad \begin{cases} m_{d_B}^{(t)}(E) = 0.4 \\ m_{d_B}^{(t)}(\overline{E}) = 0.3 \\ m_{d_B}^{(t)}(\Omega) = 0.3 \end{cases}$$

$$\begin{cases} m_{d_C}^{(t)}(E) = 0.15 \\ m_{d_C}^{(t)}(\overline{E}) = 0.7 \\ m_{d_C}^{(t)}(\Omega) = 0.15 \end{cases} \quad \begin{cases} m_{d_D}^{(t)}(E) = 0.8 \\ m_{d_D}^{(t)}(\overline{E}) = 0.02 \\ m_{d_D}^{(t)}(\Omega) = 0.18 \end{cases}$$

Les masses m_{l_i} et m_{p_i} sont initialisées à m_Ω. Nous utilisons la confiance directe pour mettre à jour la connaissance locale du noeud v_i avec la règle de Dempster. La connaissance locale est mise à jour de la manière suivante en utilisant l'équation 1.6 :

$$m_{l_i}^{(t)} = m_{l_i}^{(t-1)} \oplus m_{d_i}^{(t)}, \tag{2.3}$$

où $\oplus$ dénote la règle de Dempster. La connaissance publique est mise à jour par la connaissance locale en utilisant la règle prudente (équation 1.11) :

$$m_{p_i}^{(t)} = m_{p_i}^{(t-1)} \textcircled{\wedge} m_{l_i}^{(t)}. \tag{2.4}$$

Le choix de l'opérateur prudent est justifié par le fait que le noeud v_i peut avoir une même connaissance locale à différents pas de temps. L'idempotence de cet opérateur permet d'éviter de combiner la même information plusieurs fois.

Exemple : *A l'état initial, les noeuds n'ont aucune connaissance locale ou publique sur l'évènement. $m_{l_i}^{(t)}$ et $m_{p_i}^{(t)}$ ont les mêmes valeurs que $m_{d_i}^{(t)}$ car $m_{l_i}^{(t)} = m_\Omega \oplus m_{d_i}^{(t)} = m_{d_i}^{(t)}$ et $m_{p_i}^{(t)} = m_\Omega \textcircled{\wedge} m_{l_i}^{(t)}$* (ceci est dû à l'initialisation).

L'avis des autres nœuds est nécessaire dans les algorithmes distribués. Lorsqu'un échange de messages a lieu dans le réseau, le récepteur peut ne pas recevoir les messages (pour des raisons de connexion ou autres), ou recevoir des messages qui peuvent être certains, incertains ou erronés. Si le récepteur ne reçoit pas de messages, il met à jour sa connaissance publique par sa connaissance locale comme détaillé ci-dessus et l'envoie dans le réseau.

Exemple : *Si le noeud v_A ne reçoit pas de message, il envoie sa connaissance publique dans le réseau. Son message est incertain puisqu'il ne détecte pas l'événement E.*

Dans le cas où un noeud reçoit des messages, différents traitements sont à prévoir. Pour prendre en compte l'incertitude des messages reçus, les erreurs et la non fiabilité de l'émetteur, nous avons choisi que le récepteur affaiblisse les connaissances reçues avant de les combiner avec la connaissance interne du nœud. Le coefficient d'affaiblissement α est calculé selon l'application envisagée (ligne 9 de l'algorithme). La connaissance de l'émetteur est affaiblie selon l'équation suivante :

$$^{\alpha}m_{p_k}^{(t)} = (1-\alpha).m_{p_k}^{(t)} + \alpha.m_{\Omega}^{(t)}. \tag{2.5}$$

Le récepteur prend en compte le message reçu même s'il était erroné ou incertain ; il l'affaiblit pour ensuite le combiner avec la connaissance publique. Pour mettre à jour la connaissance publique du récepteur, nous utilisons la règle prudente [Den08] détaillée dans la section 1.3.2 du chapitre 1. Dans un système distribué, la même information peut être reçue et traitée plusieurs fois. En combinant l'information, il est utile d'utiliser une règle idempotente pour éviter les cycles de disséminations, c'est à dire éviter de prendre en considération la même information plusieurs fois (data incest) comme si elle provenait de différentes sources indépendantes ([MKE04],[Mit07]). La connaissance publique du récepteur est combinée avec celle de l'émetteur affaiblie comme suit :

$$m_{p_i}^{(t)} = m_{p_i}^{(t)} \textcircled{\scriptsize$\alpha$} \, m_{p_k}^{(t)}. \tag{2.6}$$

La connaissance publique est rediffusée à travers le réseau.

Exemple : *Considérons que les messages de l'exemple sont échangés de la manière suivante :*

- *v_B reçoit un message de v_A et de v_D,*
- *v_D reçoit un message de v_C,*
- *v_D reçoit un message de v_B qui a mis à jour sa connaissance publique.*

Nous supposons que le facteur d'affaiblissement α est une constante dans cet exemple et est égal à 0.9. v_B reçoit un message de v_D et v_A et va mettre à jour sa connaissance publique. Après avoir affaibli les messages reçus et fusionné, la connaissance publique de v_B est présentée par le tableau 2.4.

TABLE 2.4.: *Connaissance publique de v_B*

v_B initial	v_B avec v_A	v_B avec v_A puis v_D
$m_{p_B}^{(t)}(E) = 0.4$	$m_{p_B}^{(t)}(E) = 0.4$	$m_{p_B}^{(t)}(E) = 0.58$
$m_{p_B}^{(t)}(NE) = 0.3$	$m_{p_B}^{(t)}(NE) = 0.3$	$m_{p_B}^{(t)}(NE) = 0.21$
$m_{p_B}^{(t)}(\Omega) = 0.3$	$m_{p_B}^{(t)}(\Omega) = 0.3$	$m_{p_B}^{(t)}(\Omega) = 0.21$

Le tableau 2.5 montre la mise à jour de la connaissance publique de v_D après la réception du message de v_C et de v_B.

TABLE 2.5.: *Connaissance publique de v_D*

v_D initial	v_D avec v_C	v_D avec v_C puis v_B mis à jour
$m_{p_D}^{(t)}(E) = 0.8$	$m_{p_D}^{(t)}(E) = 0.54$	$m_{p_D}^{(t)}(E) = 0.545$
$m_{p_D}^{(t)}(NE) = 0.02$	$m_{p_D}^{(t)}(NE) = 0.33$	$m_{p_D}^{(t)}(NE) = 0.333$
$m_{p_D}^{(t)}(\Omega) = 0.18$	$m_{p_D}^{(t)}(\Omega) = 0.12$	$m_{p_D}^{(t)}(\Omega) = 0.122$

Cet exemple montre qu'à la réception d'un message incertain comme le message de v_A, la connaissance publique du récepteur n'est pas affectée. Si le message est erroné, le récepteur l'affaiblit avant de la combiner à sa connaissance publique. La réception de plusieurs messages erronés successifs risque de perturber la croyance mais on espère que les autres messages corrects vont renforcer la connaissance publique du récepteur.

Recalage temporel

Dans le cadre de la fusion distribuée, il faut prendre en compte le temps, à savoir :
- la fréquence d'envoi des messages,
- le temps de traitement des messages,
- le temps de calcul de la fusion distribuée.

En effet, les données contenues dans les messages ont été élaborées à un temps passé. Afin de prendre en compte l'obsolescence des données, nous effectuons un affaiblissement des masses lorsque nous les fusionnons avec des masses plus récentes. L'affaiblissement temporel est effectué en utilisant l'équation 1.18 où α dans

ce cas est remplacé par ζ :

$$^{\zeta}m^{(t)} = (1-\zeta).m^{(t)} + \zeta m_{\Omega}^{(t)} \tag{2.7}$$

où ζ est définie en fonction du temps :

$$\zeta = 1 - f(\Delta t). \tag{2.8}$$

Ainsi, dans l'algorithme 1, les formules

$$\begin{array}{l} m_{l_i}^{(t)} \leftarrow m_{l_i}^{(t-1)} \oplus m_{d_i}^{(t)}, \\ m_{p_i}^{(t)} \leftarrow m_{p_i}^{(t-1)} \oslash m_{l_i}^{(t)}, \end{array}$$

sont remplacées par :

$$\begin{array}{l} m_{l_i}^{(t)} \leftarrow^{\zeta} m_{l_i}^{(t-1)} \oplus m_{d_i}^{(t)}, \\ m_{p_i}^{(t)} \leftarrow^{\varsigma} m_{p_i}^{(t-1)} \oslash m_{l_i}^{(t)}. \end{array}$$

Le recalage temporel est représenté dans l'algorithme 2 par les symboles ζ et ς. Il faut noter que ces coefficients ζ et ς sont généralement calculés à partir de la même fonction f.

Cas de différents éléments d'observation

La connaissance échangée dans les messages peut concerner différents éléments d'observation. Par exemple, si la connaissance directe provient d'un capteur embarqué qui détecte plusieurs objets ou évènements, le noeud peut disposer d'informations sur différents objets au même instant représentés par un vecteur de masses $[m_{p_i}^{(t)}]$. Cette situation nécessite une association entre les différentes connaissances avant de les combiner. L'algorithme 2 est modifié pour prendre en compte l'association : Association(C_{locale_i}, $C_{directe_i}$) (ligne 5) pour l'association de la connaissance directe avec la connaissance locale, Association($C_{publique_i}$, C_{locale_i}) (ligne 13) pour l'association de la connaissance publique avec la connaissance locale, Association($C_{publique_i}$, $C_{publique_k}$) (ligne 23) pour l'association de la connaissance publique du récepteur avec celle de l'émetteur. Dans cette partie, nous n'allons pas détailler l'association.

Algorithme 2 : Fusion distribuée à la réception d'un message sur le noeud i : cas d'une masse sur un seul élément avec des affaiblissements temporels

Données : Message de l'émetteur v_k contenant $m_{p_k}^{(t)}$;
Résultats : C_{locale_i}=$m_{l_i}^{(t)}$ et $C_{publique_i}$=$m_{p_i}^{(t)}$;
$m_{d_i}^{(t)} \leftarrow ConfianceDirecte()$;
Mise à jour de la connaissance locale
$\zeta \leftarrow CalculAffTemp()$;
${}^{\zeta}m_{l_i}^{(t)} \leftarrow (1-\zeta).m_{l_i}^{(t)} + \zeta.m_{\Omega}^{(t)}$;
$m_{l_i}^{(t)} \leftarrow^{\zeta} m_{l_i}^{(t-1)} \oplus m_{d_i}^{(t)}$;
Mise à jour de la connaissance publique
$\varsigma \leftarrow CalculAffTemp()$;
${}^{\varsigma}m_{p_i}^{(t)} \leftarrow (1-\varsigma).m_{p_i}^{(t)} + \varsigma.m_{\Omega}^{(t)}$;
Avec la connaissance locale
$m_{p_i}^{(t)} \leftarrow^{\varsigma} m_{p_i}^{(t-1)} \textcircled{\wedge} m_{l_i}^{(t)}$;
$\alpha \leftarrow CalculAffaiblissement()$;
${}^{\alpha}m_{p_k}^{(t)} \leftarrow (1-\alpha).m_{p_k}^{(t)} + \alpha.m_{\Omega}^{(t)}$;
Avec le message
$m_{p_i}^{(t)} \leftarrow m_{p_i}^{(t)} \textcircled{\wedge}^{\alpha} m_{p_k}^{(t)}$;
Envoi message contenant $m_{p_i}^{(t)}$;

Cette étape est décrite dans le chapitre 5. Nous proposons donc une troisième version de l'algorithme, l'algorithme 3 qui fusionne les informations après l'étape d'association.

Exemple 1 : *Dans cet exemple, nous considérons le cas de deux noeuds F et G qui détectent différents objets dans la scène. Noeud F détecte deux objets O_1 et O_2. Noeud G détecte seulement un objet O_2, il n'a pas confiance en cet objet, il vient de le détecter. Cet exemple est représenté dans la figure 2.6. La confiance directe est la suivante :*

$$\begin{cases} m_{d_F}^{(t)}(1) = 0.87 \\ m_{d_F}^{(t)}(\overline{1}) = 0.03 \\ m_{d_F}^{(t)}(\Omega) = 0.1 \end{cases} \quad \begin{cases} m_{d_F}^{(t)}(2) = 0.71 \\ m_{d_F}^{(t)}(\overline{2}) = 0.18 \\ m_{d_F}^{(t)}(\Omega) = 0.1 \end{cases}$$

$$\begin{cases} m_{d_G}^{(t)}(2) = 0.08 \\ m_{d_G}^{(t)}(\overline{2}) = 0.8 \\ m_{d_G}^{(t)}(\Omega) = 0.12 \end{cases}$$

Algorithme 3 : Fusion distribuée à la réception d'un message sur le noeud i : cas de différents éléments d'observation avec recalage temporel

Données : Message de l'émetteur v_k comprenant $[m_{p_k}^{(t)}]$;
Résultats : C_{locale_i}=$[m_{l_i}^{(t)}]$ et $C_{publique_i}$=$[m_{p_i}^{(t)}]$;
$[m_{d_i}^{(t)}] \leftarrow ConfianceDirecte()$;
Mise à jour de la connaissance locale
Association(C_{locale_i}, $C_{directe_i}$);
  pour chaque composante de C_{locale_i} et $C_{directe_i}$ associées
    $m_{l_i}^{(t)} \leftarrow^{\zeta} m_{l_i}^{(t-1)} \oplus m_{d_i}^{(t)}$
  pour chaque composante de C_{locale_i} non associée
    $m_{l_i}^{(t)} \leftarrow^{\zeta} m_{l_i}^{(t-1)}$
  pour chaque composante de $C_{directe_i}$ non associée
    Ajouter cette composante à C_{locale_i}
Mise à jour de la connaissance publique
Association($C_{publique_i}$, C_{locale_i});
  pour chaque composante de C_{locale_i} et $C_{publique_i}$ associées
    $m_{p_i}^{(t)} \leftarrow^{\varsigma} m_{p_i}^{(t-1)} \oslash m_{l_i}^{(t)}$
  pour chaque composante de $C_{publique_i}$ non associée
    $m_{p_i}^{(t)} \leftarrow^{\varsigma} m_{p_i}^{(t-1)}$
  pour chaque composante de C_{locale_i} non associée
    Ajouter cette composante à $C_{publique_i}$
for *chaque message provenant de* v_k **do**
    $\alpha \leftarrow CalculAffaiblissement()$
    ${}^{\alpha}m_{p_k}^{(t)} \leftarrow (1-\alpha).m_{p_k}^{(t)} + \alpha.m_{\Omega}^{(t)}$;
    Association($C_{publique_i}$, $C_{publique_k}$);
      pour chaque composante de $C_{publique_i}$ et $C_{publique_k}$ associées
        $m_{p_i}^{(t)} \leftarrow m_{p_i}^{(t)} \oslash^{\alpha} m_{p_k}^{(t)}$
      pour chaque composante de $C_{publique_k}$ non associée
        Ajouter cette composante à $C_{publique_i}$
Envoi message contenant $[m_{p_i}^{(t)}]$;

$m_{l_F}^{(t)}$, $m_{l_G}^{(t)}$ sont initialisées à m_d. L'algorithme d'association utilisé dans cet exemple est détaillé dans le chapitre 5. Les noeuds F et G échangent leur connaissance publique par messages.

Noeud F reçoit l'objet O_2 déjà détecté dans sa connaissance locale. En associant la connaissance publique avec le message, il met à jour sa connaissance publique avec la règle prudente. La connaissance publique résultante est la suivante :

$$\begin{cases} m_{p_F}^{(t)}(1) = 0.98 \\ m_{p_F}^{(t)}(\overline{1}) = 0.01 \\ m_{p_F}^{(t)}(\Omega) = 0.01 \end{cases} \quad \begin{cases} m_{p_F}^{(t)}(2) = 0.79 \\ m_{p_F}^{(t)}(\overline{2}) = 0.14 \\ m_{pF}^{(t)}(\Omega) = 0.06 \end{cases}$$

Le noeud G reçoit O_1 et O_2 du noeud F. O_2 est déjà détecté dans sa connaissance locale. En associant la connaissance publique avec le message, O_2 détecté par F est associé à O_2 détecté par G. Il met à jour sa fonction de masse en utilisant la règle prudente. O_1 n'est pas associé, il sera ajouté à la connaissance publique du noeud G après l'avoir affaiblit. Le résultat de la mise à jour des fonctions de masse est le suivant :

$$\begin{cases} m_{p_G}^{(t)}(1) = 0.56 \\ m_{p_G}^{(t)}(\overline{1}) = 0.14 \\ m_{p_G}^{(t)}(\Omega) = 0.29 \end{cases} \quad \begin{cases} m_{p_G}^{(t)}(2) = 0.65 \\ m_{p_G}^{(t)}(\overline{2}) = 0.30 \\ m_{pG}^{(t)}(\Omega) = 0.05 \end{cases}$$

Cet exemple montre l'intérêt de l'algorithme de fusion de données distribuée. Il permet d'augmenter le champ de vision des noeuds.

2.3.3. Convergence de l'algorithme

La convergence de l'algorithme a été démontrée dans [DCD12]. Les auteurs ont démontré que l'opérateur prudent avec un affaiblissement sur les poids est un r-opérateur idempotent. Les r-opérateurs conduisent sous certaines conditions à l'auto-stabilisation du calcul global [Duc07]. Un r-opérateur peut être construit à partir d'un opérateur idempotent, associatif et commutatif. Ces conditions sont vérifiées par l'opérateur prudent. Ce dernier définit aussi une relation d'ordre $\sqsubseteq_w$ (voir section 1.3.2 chapitre 1) et se base sur le minimum des poids. L'opérateur prudent avec les affaiblissements sur les poids vérifie un endomorphisme croissant, ce qui est l'une des conditions d'un r-opérateur. En se basant sur toutes ces conditions, les auteurs ont prouvé ainsi l'auto-stabilisation de l'algorithme en utilisant l'opérateur prudent et l'affaiblissement sur les poids.

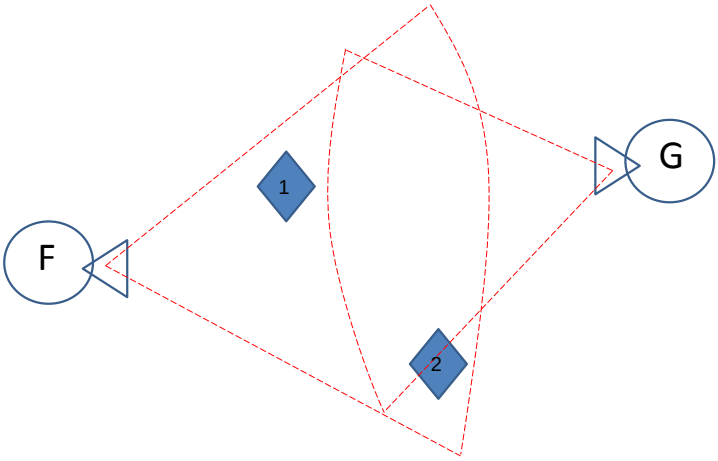


FIGURE 2.6.: *Exemple de configuration : F détecte les objets 1 et 2, G détecte l'objet 2. Les noeuds échangent des messages.*

L'algorithme de fusion distribuée proposé dans cette thèse est une adaptation puisqu'il est basé sur le calcul des masses. Nous n'avons pas démontré la convergence de l'algorithme en utilisant les masses puisque les masses et les poids sont en relation selon l'équation 1.12 de la décomposition canonique. Quand les poids augmentent les masses diminuent. L'affaiblissement des masses correspond donc bien à une augmentation des poids, nécessaire à la convergence de l'algorithme. Nous allons illustrer la convergence de l'algorithme avec deux exemples utilisant les masses. Considérons à nouveau l'exemple de la figure 2.4, en supposant cette fois-ci que les noeuds A, B, C et D envoient leurs messages à tout le réseau et que le capteur du noeud C n'est pas défectueux.

Exemple 2 : *Soit la confiance directe sur l'évènement E des quatre noeuds de l'exemple à un instant initial noté t :*

$$\begin{cases} m_{d_A}^{(t)}(E) = 0 \\ m_{d_A}^{(t)}(\overline{E}) = 0 \\ m_{d_A}^{(t)}(\Omega) = 1 \end{cases} \quad \begin{cases} m_{d_B}^{(t)}(E) = 0.7 \\ m_{d_B}^{(t)}(\overline{E}) = 0.05 \\ m_{d_B}^{(t)}(\Omega) = 0.25 \end{cases}$$

$$\begin{cases} m_{d_C}^{(t)}(E) = 0.55 \\ m_{d_C}^{(t)}(\overline{E}) = 0.3 \\ m_{d_C}^{(t)}(\Omega) = 0.15 \end{cases} \quad \begin{cases} m_{d_D}^{(t)}(E) = 0.8 \\ m_{d_D}^{(t)}(\overline{E}) = 0.02 \\ m_{d_D}^{(t)}(\Omega) = 0.18 \end{cases}$$

Les masses m_{l_i} et m_{p_i} sont initialisées à m_Ω. $m_{l_i}^{(t)}$ et $m_{p_i}^{(t)}$ ont les mêmes valeurs que $m_{d_i}^{(t)}$ car $m_{l_i}^{(t)} = m_\Omega \oplus m_{d_i}^{(t)}$ et $m_{p_i}^{(t)} = m_\Omega \textcircled{\cap} m_{l_i}^{(t)}$. Chaque noeud envoie sa connaissance publique à tous les autres noeuds du réseau. Le noeud récepteur met à jour sa connaissance publique comme décrit dans l'algorithme 3. Il faut noter que dans cet exemple nous ne faisons pas d'association, nous combinons simplement les masses avec la règle de Dempster et l'opérateur prudent. On obtient après l'échange de 17 messages le résultat suivant :

$$\begin{cases} m_{p_A}^{(t)}(E) = 0.75 \\ m_{p_A}^{(t)}(\overline{E}) = 0.15 \\ m_{p_A}^{(t)}(\Omega) = 0.01 \end{cases} \quad \begin{cases} m_{p_B}^{(t)}(E) = 0.98 \\ m_{p_B}^{(t)}(\overline{E}) = 0.01 \\ m_{p_B}^{(t)}(\Omega) = 0.01 \end{cases}$$

$$\begin{cases} m_{p_C}^{(t)}(E) = 0.98 \\ m_{p_C}^{(t)}(\overline{E}) = 0.01 \\ m_{p_C}^{(t)}(\Omega) = 0.01 \end{cases} \quad \begin{cases} m_{p_D}^{(t)}(E) = 0.98 \\ m_{p_D}^{(t)}(\overline{E}) = 0.01 \\ m_{p_D}^{(t)}(\Omega) = 0.01 \end{cases}$$

Les masses des noeuds A, B, C et D convergent vers l'existence de l'évènement E. Cependant, le noeud A ne converge pas vers le même résultat : il est influencé par sa connaissance directe car il ne détecte pas l'évènement.

Exemple 3 : *Nous reprenons la configuration précédente dans laquelle les noeuds A, B C et D envoient leur message à tout le réseau, mais au dixième message (et seulement au dixième), le capteur du noeud C donne des informations erronées. Après cinq itérations pour chaque noeud, le résultat est le suivant :* **à t=5,**

$$\begin{cases} m_{p_A}^{(t)}(E) &= 0.75 \\ m_{p_A}^{(t)}(\overline{E}) &= 0.15 \\ m_{p_A}^{(t)}(\Omega) &= 0.1 \end{cases} \quad \begin{cases} m_{p_B}^{(t)}(E) &= 0.97 \\ m_{p_B}^{(t)}(\overline{E}) &= 0.02 \\ m_{p_B}^{(t)}(\Omega) &= 0.01 \end{cases}$$

$$\begin{cases} m_{p_C}^{(t)}(E) &= 0.85 \\ m_{p_C}^{(t)}(\overline{E}) &= 0.13 \\ m_{p_C}^{(t)}(\Omega) &= 0.02 \end{cases} \quad \begin{cases} m_{p_D}^{(t)}(E) &= 0.97 \\ m_{p_D}^{(t)}(\overline{E}) &= 0.02 \\ m_{p_D}^{(t)}(\Omega) &= 0.01 \end{cases}$$

Après l'échange de neuf messages, les avis des noeuds sur l'évènement E sont les suivants :

à t=9,

$$\begin{cases} m_{p_A}^{(t)}(E) &= 0.754 \\ m_{p_A}^{(t)}(\overline{E}) &= 0.16 \\ m_{p_A}^{(t)}(\Omega) &= 0.086 \end{cases} \quad \begin{cases} m_{p_B}^{(t)}(E) &= 0.98 \\ m_{p_B}^{(t)}(\overline{E}) &= 0.01 \\ m_{p_B}^{(t)}(\Omega) &= 0.01 \end{cases}$$

$$\begin{cases} m_{p_C}^{(t)}(E) &= 0.95 \\ m_{p_C}^{(t)}(\overline{E}) &= 0.03 \\ m_{p_C}^{(t)}(\Omega) &= 0.02 \end{cases} \quad \begin{cases} m_{p_D}^{(t)}(E) &= 0.98 \\ m_{p_D}^{(t)}(\overline{E}) &= 0.01 \\ m_{p_D}^{(t)}(\Omega) &= 0.01 \end{cases}$$

Au dixième message, le capteur du noeud C donne des informations erronées. La confiance directe de C au dixième message est :

$$\begin{cases} m_{d_C}^{(t)}(E) &= 0.15 \\ m_{d_C}^{(t)}(\overline{E}) &= 0.7 \\ m_{d_C}^{(t)}(\Omega) &= 0.15 \end{cases}$$

Après combinaison, les avis des noeuds sur l'évènement E sont :

à t=10,

$$\begin{cases} m_{p_A}^{(t)}(E) &= 0.73 \\ m_{p_A}^{(t)}(\overline{E}) &= 0.18 \\ m_{p_A}^{(t)}(\Omega) &= 0.09 \end{cases} \quad \begin{cases} m_{p_B}^{(t)}(E) &= 0.98 \\ m_{p_B}^{(t)}(\overline{E}) &= 0.01 \\ m_{p_B}^{(t)}(\Omega) &= 0.01 \end{cases}$$

$$\begin{cases} m_{p_C}^{(t)}(E) &= 0.90 \\ m_{p_C}^{(t)}(\overline{E}) &= 0.08 \\ m_{p_C}^{(t)}(\Omega) &= 0.02 \end{cases} \quad \begin{cases} m_{p_D}^{(t)}(E) &= 0.97 \\ m_{p_D}^{(t)}(\overline{E}) &= 0.02 \\ m_{p_D}^{(t)}(\Omega) &= 0.01 \end{cases}$$

Le capteur C donne de nouveau une information correcte. Après plusieurs échanges de messages, les avis convergent vers :

$$\begin{cases} m^{(t)}_{p_A}(E) = 0.75 \\ m^{(t)}_{p_A}(\overline{E}) = 0.15 \\ m^{(t)}_{p_A}(\Omega) = 0.01 \end{cases} \quad \begin{cases} m^{(t)}_{p_B}(E) = 0.98 \\ m^{(t)}_{p_B}(\overline{E}) = 0.01 \\ m^{(t)}_{p_B}(\Omega) = 0.01 \end{cases}$$

$$\begin{cases} m^{(t)}_{p_C}(E) = 0.98 \\ m^{(t)}_{p_C}(\overline{E}) = 0.01 \\ m^{(t)}_{p_C}(\Omega) = 0.01 \end{cases} \quad \begin{cases} m^{(t)}_{p_D}(E) = 0.98 \\ m^{(t)}_{p_D}(\overline{E}) = 0.01 \\ m^{(t)}_{p_D}(\Omega) = 0.01 \end{cases}$$

Cet exemple illustre le fait que l'algorithme de fusion distribué converge en utilisant l'opérateur prudent avec affaiblissement des masses. L'exemple 3 converge vers le même résultat de l'exemple 2, ce qui montre que l'algorithme est tolérant aux fautes.

2.4. Conclusion

Nous avons décrit dans ce chapitre un algorithme de fusion distribuée adapté aux réseaux ad hoc mobiles. Chaque noeud met à jour une connaissance locale et une connaissance distribuée avec des opérateurs appropriés (opérateur de Dempster, combinaison prudente, affaiblissement) prenant en compte les cycles possibles dans un tel réseau. Le formalisme des fonctions de croyance est utilisé pour représenter les incertitudes. L'algorithme peut être adapté au cas d'étude en considérant un ou plusieurs éléments d'observation et en prenant en compte l'obsolescence des données grâce à l'affaiblissement temporel. Cet algorithme de fusion distribuée a été utilisé pour le développement de deux applications : la détection de faux véhicules dans une attaque "sybil" et la perception augmentée de véhicules. Ces applications sont détaillées dans les chapitres 3 et 4.

CHAPITRE 3

Détection de faux véhicules dans une attaque "sybil"

La validation de l'algorithme de fusion de données distribuées a été faite en premier lieu sur une application de détection de l'attaque sybil dans un réseau de véhicules. Nous présentons dans ce chapitre l'approche développée ainsi que les résultats. Il s'agit d'établir pour chaque noeud la confiance qu'il peut placer dans chacun des autres noeuds du réseau. Les noeuds diffusent leurs avis sur l'ensemble des noeuds qui sont réutilisés à la réception pour évaluer d'autres noeuds.

3.1. L'attaque *Sybil* : Définition et état de l'art

L'attaque *Sybil*, présentée par Douceur [Dou02], constitue une menace grave, car elle attaque la fonctionnalité des VANETs. Au moment de l'attaque, un noeud attaquant, appelé *malveillant* est capable de revendiquer des entités multiples appelées *noeuds sybils* ou *faux noeuds*. Il envoie des messages avec des identités multiples aux autres noeuds du réseau. Ainsi, en se faisant passer pour ces différentes identités, le noeud malveillant pourra compromettre plus facilement le fonctionnement général du réseau de véhicules. L'attaque peut consister à donner l'illusion d'un embouteillage ou d'accident afin que les autres véhicules changent de chemin ou quittent la route pour le bénéfice de l'attaquant. Le noeud malveillant peut également injecter de fausses informations dans les réseaux via de faux noeuds fabriqués. Pour plus d'informations sur les différentes attaques possibles dans le réseau de véhicules, le

lecteur peut consulter l'annexe A.

Définition 0 Un noeud est une entité physique ayant une identité, une position. Il peut recevoir et émettre des messages.

Définition 1 Un noeud *malveillant* (malicious node, attack node) est un noeud qui envoie des messages destinés à tromper les autres noeuds du réseau. La position de l'émetteur dans le message est modifiée afin de faire croire à l'existence de faux noeuds.

Définition 2 Un *faux* noeud (fake node, sybil node) est un noeud qui n'existe pas physiquement dans le réseau. Il est créé par un noeud malveillant afin de perturber l'analyse de la situation (simulation d'embouteillage, annonce de fausses informations en nombre, etc.).

Les attaques *Sybil* peuvent être classées en trois catégories selon le type de communication, d'identité, et leur participation dans le réseau. Ces catégories sont brièvement discutées dans les paragraphes suivants.

1. Communication : Quand un noeud honnête envoie un message à un noeud Sybil, le noeud malveillant écoute le message. De la même manière, les messages envoyés à partir des noeuds Sybil sont en fait envoyés par l'un des noeuds malveillants.
2. Identité : Lors d'une attaque Sybil, un attaquant crée une nouvelle identité Sybil. Un attaquant peut fabriquer cette identité (fabrication d'identité) ou il peut usurper l'identité légitime d'un de ses voisins (vol d'identité).
3. Participation : de multiples identités Sybil créées par des noeuds malveillants peuvent participer simultanément à l'attaque ou l'attaquant peut utiliser ces identités une par une. Une identité particulière peut quitter ou rejoindre le réseau à plusieurs reprises.

Une attaque *Sybil* peut nuire au fonctionnement de tout le réseau. On peut citer différents cas :

– L'agrégation des données : Grâce à de multiples identités, un noeud malveillant peut contribuer à la prise en compte plusieurs fois de la même donnée comme s'il s'agissait d'informations différentes et modifier ainsi le résultat de l'agréga-

tion des données.

- Routage : Les attaques sybil sont efficaces contre le fonctionnement des protocoles de routage des VANETs. Dans un routage multichemins, différents chemins sont utilisés. La présence des identités Sybil d'un noeud malveillant sur ces chemins peut nuire à l'acheminement. Le routage géographique est également vulnérable puisqu'un noeud malveillant peut apparaître dans plusieurs endroits à la fois.
- Vote : L'attaque Sybil peut mettre à jour le résultat du vote des noeuds d'une manière incorrecte. Si l'attaquant crée suffisamment de faux noeuds participant à l'évaluation du comportement d'un noeud, un vrai noeud peut être expulsé du réseau suite à ce vote.
- Détection d'un comportement suspect (misbehavior) : Un attaquant peut éviter d'être détecté en répartissant la responsabilité sur les faux noeuds. Si le mécanisme de détection utilise de multiples observations de localisation du noeud malveillant, l'attaquant peut toujours échapper à la détection en utilisant des noeuds différents à des moments différents. Si certains noeuds Sybil sont détectés et expulsés du réseau pour les comportements malveillants, l'attaquant peut utiliser d'autres identités.

Nous cherchons à quantifier la confiance dans un noeud du réseau afin de détecter l'attaque sybil dans un réseau de véhicules. Différentes techniques ont été développées pour détecter les faux noeuds dans les VANETs. Gole et al. [GGS04] ont proposé une méthode basée sur un principe de parcimonie qui consiste à trouver la meilleure explication pour les données corrompues. Les véhicules distinguent leurs voisins en utilisant des caméras ou en échangeant des messages dans le spectre infrarouge. La technique décrite par Xiao et al. [XBG06] pour détecter les noeuds sybils est basée sur l'analyse de la puissance du signal en utilisant comme support les infrastructures routières. Yan et al. [YCWO08] utilisent le radar pour détecter les voisins et vérifier les positions annoncées. Yu et al. [YXX13] proposent une méthode coopérative pour la détection de l'attaque sybil. Les noeuds voisins coopèrent pour mesurer la puissance du signal d'un noeud suspect et vérifier sa position physique. Piro et al. [PSL06] présentent une détection passive de l'attaque sybil en utilisant des observateurs (unique ou multiples). Compte tenu de la dynamique du réseau de véhicules, du nombre des véhicules et de la difficulté d'avoir un accès permanent aux infrastructures, les outils classiques comme le PKI (Public key infrastructure) ne

sont pas adaptés. Comme il a été montré dans [GD07], par une simple comparaison de la puissance du signal reçu, la moitié des véhicules peut détecter les faux noeuds et il est prévu que les techniques coopératives permettent de diminuer le nombre des véhicules malveillants. Un algorithme coopératif entre les véhicules permettrait d'éviter les méthodes cryptographiques.

Dans ce chapitre, nous décrivons le système et la représentation de la confiance par des fonctions de masse. Nous présentons l'algorithme proposé pour la détection de l'attaque sybil. Cette approche a été validée sur des données simulées.

3.2. Formalisation du problème de l'attaque sybil

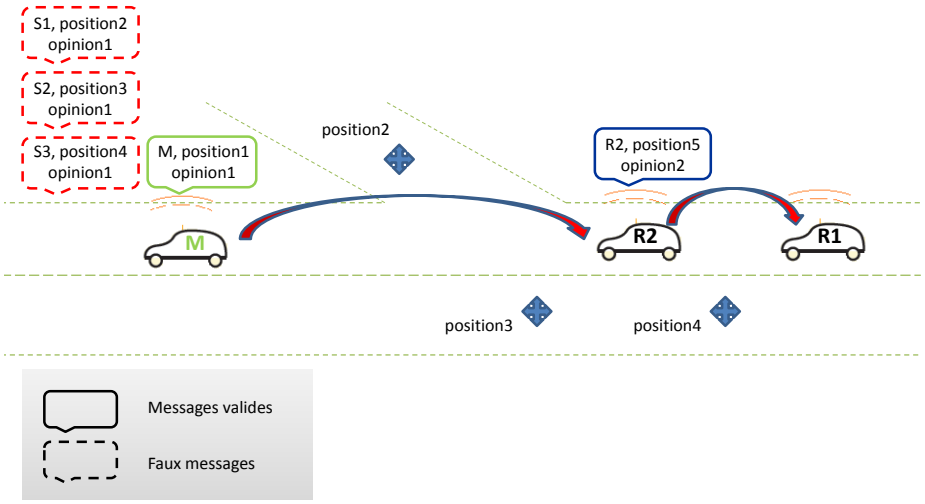
Nous considérons un réseau constitué de noeuds échangeant des messages. Chaque noeud envoie périodiquement des *messages valides* contenant son identité et sa position géographique. Le noeud malveillant envoie des *messages valides* et des *faux messages* qui contiennent une fausse identité et une fausse position. En recevant les faux messages, les autres noeuds sont leurrés et considèrent des noeuds non existants, appelés *noeuds sybil* ou *faux noeuds*. La figure 3.1 montre un exemple d'attaque sybil.

Nous adoptons les hypothèses suivantes :

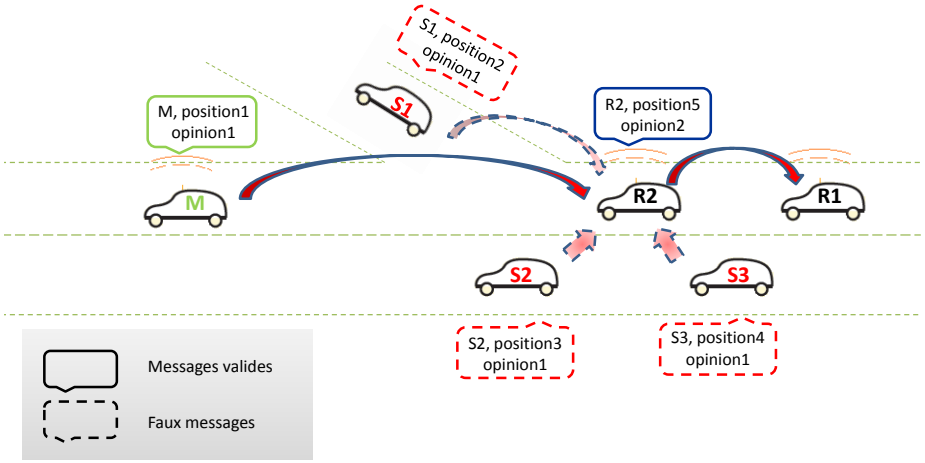
- un seul noeud malveillant crée différents noeuds sybil ;
- tous les noeuds utilisent le même système de transmission (même antenne et même puissance) ;
- la position des noeuds est supposée connue de manière précise ;
- les identités des noeuds sont connues.

La topologie du réseau est donnée par la gamme de transmission radio des noeuds (unit disk graph). Le message contient aussi la confiance de l'émetteur dans les noeuds du réseau.

Nous proposons une adaptation de l'algorithme de fusion distribuée pour combiner les données échangées dans le réseau ad hoc mobile dans le but de quantifier la confiance dans les noeuds du réseau.



(a) La scène où un noeud malveillant noté M envoie des messages au véhicule R_2 : un valide et trois faux messages.



(b) La scène telle que le véhicule R_2 est leurré en recevant le message de M.

FIGURE 3.1.: *Exemple de l'attaque sybil : M envoie 4 messages à R_2 dont 1 valide et 3 faux messages contenant 3 différentes identités et positions et le même avis que M (figure 3.1a). En recevant les messages, R_2 croit qu'il y a 4 véhicules S_1, S_2, S_3 et M (figure 3.1b).*

3.3. Représentation des connaissances échangées dans le réseau

3.3.1. Confiance dans un noeud

Chaque noeud est capable d'attribuer une confiance sur l'existence réelle de chacun des autres noeuds du réseau. Cette confiance est représentée par une masse notée m, répartie sur le cadre de discernement $\Omega = \{0, 1\}$ où 0 représente l'hypothèse que le noeud est faux et 1 l'hypothèse qu'il est vrai. Soit m_{ij} la masse correspondante qui représente l'avis d'un noeud v_i sur le noeud v_j. m_{ij} est définie sur Ω de la façon suivante :

$$\begin{aligned} m_{ij}(\emptyset) &= 0, \\ m_{ij}(0) &= p_{ij}, \\ m_{ij}(1) &= q_{ij}, \\ m_{ij}(\Omega) &= 1 - p_{ij} - q_{ij}. \end{aligned} \tag{3.1}$$

Dans cette approche, les fonctions des masses sont représentées par un vecteur de masse puisqu'il s'agit de plusieurs éléments d'observations.

3.3.2. Constitution des messages

Le noeud v_k envoie au noeud v_i un message contenant son identité, ses coordonnées et son avis sur l'ensemble des noeuds du réseau. La figure 3.2 représente les différents messages envoyés par un vrai noeud Id_V et un noeud malveillant Id_M et Id_S. On peut remarquer que le faux message avec l'identité Id_S a le même avis sur l'ensemble du réseau que les messages valides envoyés par le noeud malveillant.

A la réception, le noeud v_i établit, après avoir analysé la puissance du signal, une confiance directe de lui même v_i sur le noeud v_k. Celle-ci est représentée par une masse noté $m_{d_{ik}}$. Cette confiance est conservée dans la *connaissance locale*.

Il faut rappeler que chaque noeud possède deux types de connaissance : *locale* et *publique ou distribuée*. La connaissance locale dépend seulement de la puissance du signal des messages et non de leurs contenus : par conséquent, elle ne peut pas être influencée par les avis des autres noeuds. En revanche, la connaissance publique est le résultat de la combinaison des contenus des messages et peut être influencée

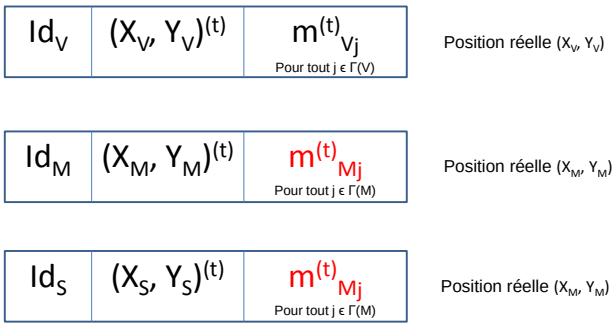


FIGURE 3.2.: *Messages valides et faux : le premier représente un message valide d'un vrai noeud. Le deuxième est un message valide envoyé par le noeud malveillant. Le troisième représente un faux message envoyé par le noeud malveillant sous l'identité* Id_S.

par l'avis des faux noeuds. La mémoire interne de chaque noeud est représentée par deux vecteurs de masse (tableau de $|V|$ cases initialisé à m_Ω si $i \neq j$ et $m(1) = 1$ si $i = j$) :

$$\begin{aligned} C_{locale_i}(t) &= [m^{(t)}_{l_{ij}}] \\ C_{publique_i}(t) &= [m^{(t)}_{p_{ij}}]. \end{aligned} \tag{3.2}$$

3.4. Algorithme de fusion distribuée pour la détection de faux noeuds

L'algorithme 4 est utilisé dans ce cas pour traiter le message reçu par le noeud v_i afin de détecter les faux noeuds dans le réseau. Cet algorithme prend comme données le message de v_k à v_i et la puissance du signal P. Le message contient la connaissance publique du noeud v_k (son avis sur les autres noeuds) ainsi que son identité et sa position (figure 3.2).

Remarquons que cette application ne nécessite ni d'association d'objets ni synchronisation. Comme les identifiants des émetteurs sont utilisés pour le traitement des messages, il n'est pas utile d'ajouter une étape d'association pour savoir de quels

Algorithme 4 : Fusion distribuée à la réception d'un message : cas d'une attaque sybil

Données : Message de v_k ; P_r puissance reçue ;
Résultats : C_{locale_i}=$[m_{l_i}^{(t)}]$ et $C_{publique_i}$=$[m_{p_i}^{(t)}]$;
$m_{d_i} \leftarrow ConfianceDirecte(message, P_r)$;
Mise à jour de la connaissance locale
$m_{l_i}^{(t)} \leftarrow m_{l_i}^{(t)} \oplus m_{d_i}^{(t)}$;
Mise à jour de la connaissance publique
$m_{p_i}^{(t)} \leftarrow m_{p_i}^{(t-1)} \textcircled{\wedge} m_{l_i}^{(t)}$;
for *chaque message provenant de* v_k **do**
    $\alpha \leftarrow CalculAffaiblissement(m_{l_i}^{(t)})$;
    ${}^{\alpha}m_{p_k}^{(t)} \leftarrow (1-\alpha).m_{p_k}^{(t)} + \alpha.m_{\Omega}^{(t)}$;
    $m_{p_i}^{(t)} \leftarrow m_{p_i}^{(t-1)} \textcircled{\wedge}^{\alpha} m_{p_k}^{(t)}$;
Envoi message $[m_{p_i}^{(t)}]$;

noeuds il s'agit. Par ailleurs, le fait que le noeud soit vrai ou faux est constant et continu dans le temps. L'ancienneté des messages n'est donc pas importante ici. Dans cet algorithme, nous ne gérons pas les voisinages des noeuds. Les étapes de mise à jour des connaissances locales et publiques sont appliquées comme décrit dans l'algorithme 3 du chapitre 2. L'algorithme 4 est une adaptation de l'algorithme 3 sans association. Dans ce qui suit, nous allons détailler les fonctions *ConfianceDirecte*(message,P_r) et *CalculAffaiblissement*($m_{l_i}^{(t)}$) spécifiques à l'attaque sybil.

3.4.1. Confiance Directe

Différentes méthodes peuvent être utilisées pour calculer la confiance directe $m_{d_{ik}}$. Nous proposons une méthode qui permet de convertir une mesure réelle en une fonction de masse. A chaque réception de message provenant d'un émetteur, nous supposons que le récepteur peut analyser le signal reçu et détecter les incohérences physiques. Il s'agit donc de mesurer la puissance du signal reçu et de calculer la puissance théorique à partir des coordonnées de l'émetteur. La puissance estimée μ est calculée selon la formule de Friis de la façon suivante :

$$\mu = P_e.G_{SR}.\frac{1}{d_{ik}^2} \tag{3.3}$$

où

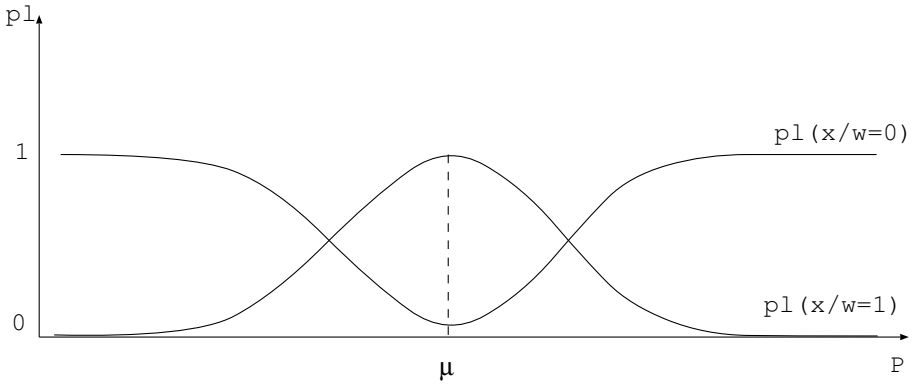


FIGURE 3.3.: *Valeurs de la plausibilité de la puissance reçue pour les vrais ($\omega = 1$) et faux ($\omega = 0$) noeuds*

- P_e est la puissance du signal émis par le noeud émetteur. Cette puissance a une valeur de référence définie selon les antennes utilisées ;
- $G_{SR} = \frac{G_e.G_r.\lambda^2}{16.\pi^2}$ est le gain, G_e et G_r étant les gains linéaires de l'antenne d'émission et de réception et λ la longueur d'ondes ;
- d_{ik} est la distance entre le noeud supposé émetteur v_k et le noeud récepteur v_i.

La comparaison entre les puissances mesurée P_r et estimée μ nous permet de détecter d'éventuels faux noeuds. La plausibilité que la puissance du signal reçu P_r soit égale à x, sachant que le noeud émetteur est un vrai noeud $(w = 1)$ est calculée de la manière suivante :

$$pl(P_r = x/\omega = 1) = \frac{f(x/\omega = 1)}{\sup_{x' \in \mathbb{R}}(f(x'/\omega = 1)} \ , \tag{3.4}$$

où $f(x/\omega = 1)$ est une fonction de densité normale de moyenne μ et de variance σ dépendant de l'antenne du récepteur.
La plausibilité $pl(P_r = x/\omega = 0)$ est définie dans l'équation suivante :

$$pl(P_r = x/\omega = 0) = 1 - k.\frac{f(x/\omega = 1)}{\sup_{x' \in \mathbb{R}}(f(x'/\omega = 1)}. \tag{3.5}$$

Elle est illustrée sur la figure 3.3 : si les puissances estimée et théorique sont égales, on conserve la possibilité que l'émetteur soit un faux noeud. En effet, si l'émetteur est un faux noeud mais si sa position est proche de celle du noeud malveillant, la position estimée est approximativement égale à la puissance mesurée. Ce résultat peut influencer la détection des faux noeuds.

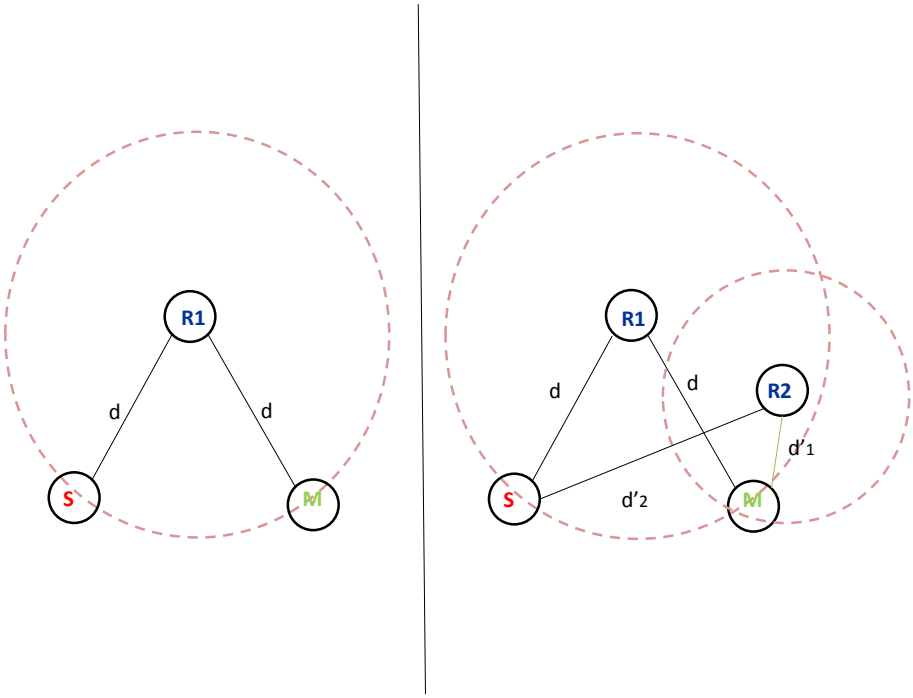


FIGURE 3.4.: *Intérêt de l'approche distribuée : influence de l'avis des autres véhicules sur la détection des noeuds sybil.*

La confiance directe est calculée en utilisant le théorème de Bayes Généralisé [Sme93b]. Elle s'obtient par la formule suivante dérivée de l'équation 1.17 :

$$\begin{aligned} m_{d_{ik}}^{(t)} &= m^{\Omega}(./P_r = x) \\ &= m_0^{\Omega} \textcircled{\cap} \{0\}^{pl(x/w=1)} \textcircled{\cap} \{1\}^{pl(x/w=0)}, \end{aligned} \tag{3.6}$$

où $\textcircled{\cap}$ représente la règle de Dempster non normalisée, m_0^{Ω} est la connaissance a priori et la notation $\{w\}^c$ représente la fonction de masse simple affectant la masse c à Ω et $1-c$ à $\{w\}$.

La confiance directe n'est pas toujours suffisante pour la détection des noeuds sybil. Pour cela, on utilise les avis des noeuds du réseau pour améliorer la détection des noeuds sybil. La figure 3.4 côté gauche présente un exemple où le noeud malveillant M a créé un noeud sybil S à la même distance que lui de R_1. Le noeud R_1 n'arrive pas à détecter que le noeud S est un noeud sybil. En recevant un message du noeud R_2 (figure 3.4 côté droit) qui lui a détecté une incohérence entre la puissance reçue et la puissance estimée, le noeud R_1 peut intégrer l'information que S est un noeud sybil.

3.4.2. Calcul du coefficient d'affaiblissement

Etant donné que l'émetteur du message n'est pas nécessairement fiable, nous avons choisi d'affaiblir ses connaissances avant de les combiner avec la connaissance interne du noeud récepteur. Le coefficient d'affaiblissement α est calculé en fonction de la connaissance locale $m_{l_{ik}}$ du noeud récepteur v_i sur le noeud émetteur v_k. Ce coefficient est égal à la plausibilité que le noeud émetteur soit non fiable :

$$\alpha = 1 - m_{l_{ik}}(1). \tag{3.7}$$

3.5. Résultats sur des données simulées

Pour valider notre approche, l'algorithme 4 a été implémenté en Matlab. Les simulations ont été faites sur des réseaux statiques et dynamiques. Pour simplifier l'analyse, nous avons d'abord supposé que les noeuds sont statiques. Nous avons effectué des simulations sur différentes configurations aléatoires du réseau. L'algorithme a ensuite été testé sur un réseau dynamique, où les noeuds sont en mouvement dans la même direction, comme des véhicules se déplaçant sur une autoroute.

3.5.1. Implémentation

Pour illustrer le fonctionnement de l'algorithme, un exemple de réseau composé de vrais noeuds VN, parmi lesquels un noeud malveillant crée différents faux noeuds FNest présenté. La puissance du signal de transmission P_e pour chacun des noeuds est égale à 600 mW et la portée de l'antenne est de l'ordre de 400m. Ces données correspondent à l'antenne D-link utilisée pour la communication entre véhicules au sein du laboratoire Heudiasyc. Nous supposons que chaque émetteur envoie son *identité*, sa *position* et sa *connaissance publique*. L'émetteur du message est choisi d'une façon aléatoire. Le récepteur utilise les informations reçues pour faire tous les calculs et vérifier si le noeud est vrai ou faux. Ceci correspond à une itération de l'algorithme. Les simulations sont effectuées jusqu'à la convergence de l'algorithme. On considère que l'algorithme converge quand $|m_{ij}^{(t-1)} - m_{ij}^{(t)}| < \epsilon$, où ϵ est un seuil prédéfini $\forall$ i,j. Les résultats de la simulation sont représentés sous forme matricielle. La figure 3.5 représente une initialisation de ces matrices. Les matrices de la première ligne correspondent à la connaissance locale et celles de la seconde

ligne à la connaissance publique. Les matrices de gauche représentent $A = 0$, du milieu $A = 1$ et de droite $A = \Omega$. Chaque case représente $m_{l_{ij}}(A)$ (première ligne), $m_{p_{ij}}(A)$ (deuxième ligne). La couleur blanche correspond à une masse égale à 1 et la noire correspond à une masse égale à 0 3.6. Les matrices sont initialisées en supposant que les vrais noeuds (ligne 1,2,4,5 et 6) ont confiance en eux mêmes. Le noeud malveillant (ligne 3) a confiance en lui et en les noeuds sybil. Les noeuds sybil (ligne 7, 8 et 9) ont le même avis que le noeud malveillant.

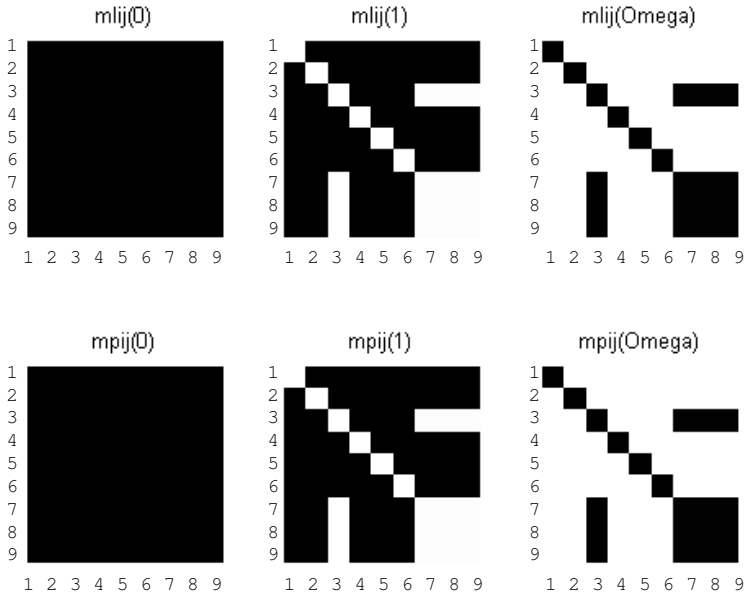


FIGURE 3.5.: *Initialisation des matrices : Les matrices de la première ligne correspondent à la connaissance locale et celles de la seconde ligne à la connaissance publique. Les matrices de gauche représentent $A = 0$, du milieu $A = 1$ et de droite $A = \Omega$. Chaque case représente $m_{l_{ij}}(A)$ (première ligne), $m_{p_{ij}}(A)$ (deuxième ligne).*

3.5.2. Réseau statique

La figure 3.7a présente un exemple de configuration du réseau (VN=6 et FN=3) où les noeuds sont statiques ; la figure 3.7b montre l'avancement de la simulation ainsi que le changement du niveau de gris à l'itération 25 ; la figure 3.7c montre

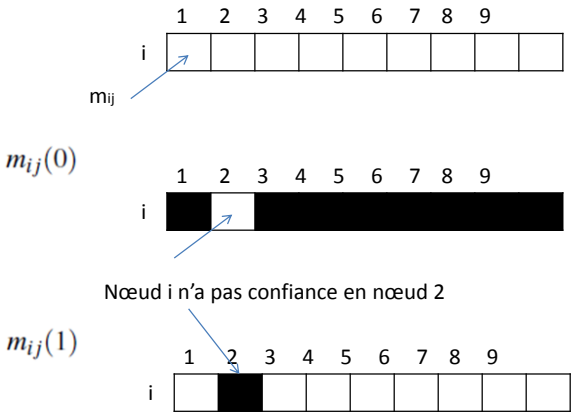


FIGURE 3.6.: *Grossissement d'une ligne de matrice : La couleur blanche correspond à une masse égale à 1 et la noire correspond à une masse égale à 0.*

le résultat de la simulation après 98 itérations. Le noeud malveillant 3 tente de convaincre les autres noeuds que les faux noeuds (7,8,9) sont de vrais noeuds. Les faux noeuds ont le même avis que le noeud malveillant. La première partie de la figure 3.7b représente la connaissance privée. Chaque noeud dispose seulement d'informations sur ses voisins. La seconde partie représente la connaissance publique. On voit que $m_{p_{ij}}(1) = 0$ pour $i = \{1, 2, 4, 5, 6\}$ et $j = \{7, 8, 9\}$, ce qui signifie que les vrais noeuds ont détecté que les noeuds $\{7, 8, 9\}$ sont des noeuds sybil.

Pour vérifier la convergence de cet algorithme, des simulations ont été faites sur différentes configurations aléatoires du réseau en changeant le nombre de faux noeuds. Le tableau 3.1 montre les résultats avec différentes proportions de faux noeuds. Une itération représente le traitement d'un message (analyse de puissance, combinaison des masses). L'algorithme prend plus de temps pour converger quand le nombre de faux noeuds augmente. Notre approche permet de détecter les noeuds sybil dans différentes configurations statiques.

TABLE 3.1.: *Résultats pour un réseau statique dans différentes configurations de réseau.*

noeuds	Moyenne des nombres d'itérations[a]	Variance du nombre d'itérations
VN=6 FN=3 [b]	207.05	7.86
VN=6 FN=4	227.55	6.89
VN=6 FN=5	255.8	6.33
VN=6 FN=6	304.7	7.55

[a] Ces résultats représentent la moyenne de 20 simulations.
[b] VN (Vrais noeuds) et FN (Faux noeuds).

3.5.3. Réseau dynamique

Les configurations statiques sont limitées, surtout dans le cas où le noeud malveillant n'est pas dans le voisinage des faux noeuds : dans cette situation, les faux noeuds ne peuvent pas être détectés. Pour cela, nous avons simulé un scénario dynamique plus réaliste où les noeuds évoluent dans la même direction comme sur une autoroute. En se déplaçant, le voisinage de chaque noeud change, ce qui influence la connaissance privée car celle-ci dépend du voisinage. Grâce à la connaissance publique, chaque noeud a accès à l'information sur tout le réseau et peut quantifier la confiance. Le tableau 3.2 montre les résultats dans différentes configurations d'un réseau dynamique. Le nombre d'itérations jusqu'à la convergence change à chaque simulation, ce qui est dû au déplacement des noeuds et au changement du voisinage. Ces résultats préliminaires montrent que les vrais noeuds peuvent détecter les faux noeuds en se déplaçant sur une autoroute.

3.6. Conclusion

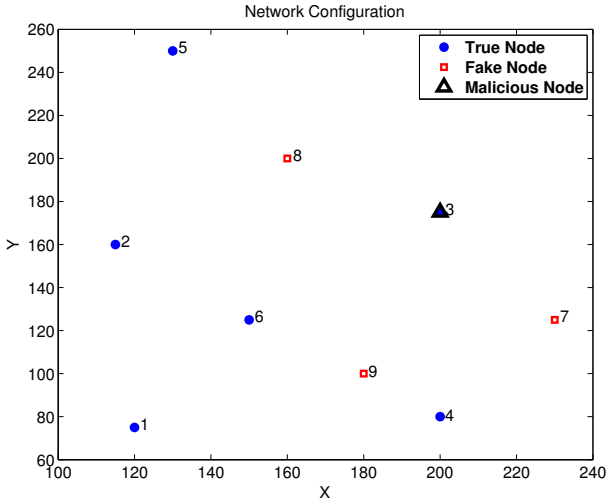
Dans ce chapitre, nous avons proposé une adaptation de l'algorithme de fusion distribuée au problème de détection de faux noeuds dans une attaque sybil dans un réseau de véhicules. La méthode présentée calcule la confiance dans un noeud sans prendre en considération le contenu des messages échangés dans le réseau. Cette méthode a été validée sur des données simulées en faisant varier le nombre

TABLE *3.2.: Résultats pour un réseau dynamique dans différentes configurations.*

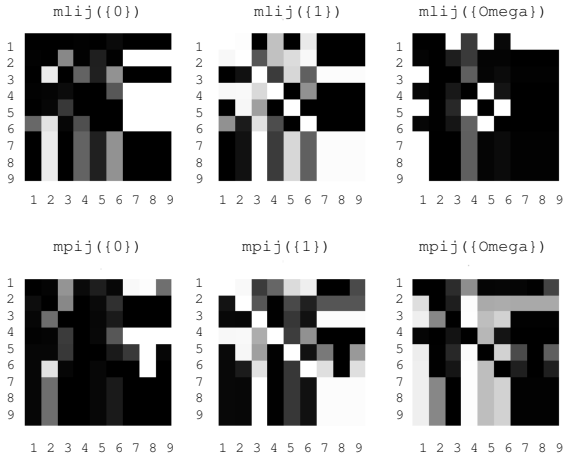
noeuds	Moyenne des nombres d'itérations[a]	Variance du nombre d'itérations
VN=6 FN=3 [b]	119.3	45.88
VN=6 FN=4	274.4	40.96
VN=6 FN=5	361.1	54.23
VN=6 FN=6	376.3	32.05

[a] Ces résultats représentent la moyenne de 10 simulations.
[b] VN (Vrais noeuds) et FN (Faux noeuds).

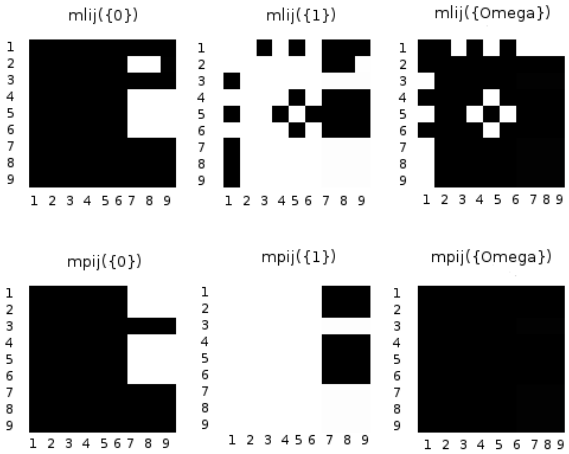
de vrais noeuds et de faux noeuds dans le réseau. Les résultats sont encourageants puisque la connaissance des noeuds sur l'état (vrai ou faux) des autres noeuds converge en un temps réaliste. Malheureusement la réalisation en situation réelle est difficile. Ceci est dû en particulier à la difficulté de mise en oeuvre du calcul de la confiance directe à l'aide de la puissance mesurée qui nécessite d'avoir des antennes identiques.



(a) Exemple de configuration géométrique du réseau.



(b) Avancement d'une simulation pour la configuration de la figure 3.7a : itération 25.



(c) Résultats d'une simulation pour la configuration de la figure 3.7a : itération 98.

FIGURE 3.7.: *Résultats de simulation d'une attaque sybil.*

CHAPITRE 4

Perception augmentée de véhicules : Carte Dynamique Distribuée

Afin de montrer l'intérêt de la fusion de données échangées dans un réseau de véhicules, nous avons développé une application de perception coopérative. Cette application appelée "Carte dynamique distribuée" consiste à cartographier les objets mobiles de la scène grâce à l'échange d'informations entre les véhicules. On augmente ainsi le champ de perception tout en diminuant les occultations. Dans cette application, la fusion distribuée devrait permettre de renforcer la croyance dans les éléments perçus et de diminuer les fausses alarmes.

4.1. Carte dynamique distribuée : définition et état de l'art

La Carte Locale Dynamique (CLD) est une carte contenant des éléments d'informations statiques et dynamiques dans le voisinage du mobile. Ces informations sont mises à jour régulièrement et en temps-réel par la perception.

Le concept de carte locale dynamique développé dans le projet Safespot [saf] et appelé LDM (Local Dynamic Map) est une bonne hiérarchisation des cartes pour l'exploitation dans le domaine des véhicules intelligents. Le concept de LDM est illustré dans la figure 4.1. La LDM contient quatre niveaux du plus statique (bas) au plus dynamique (haut) :

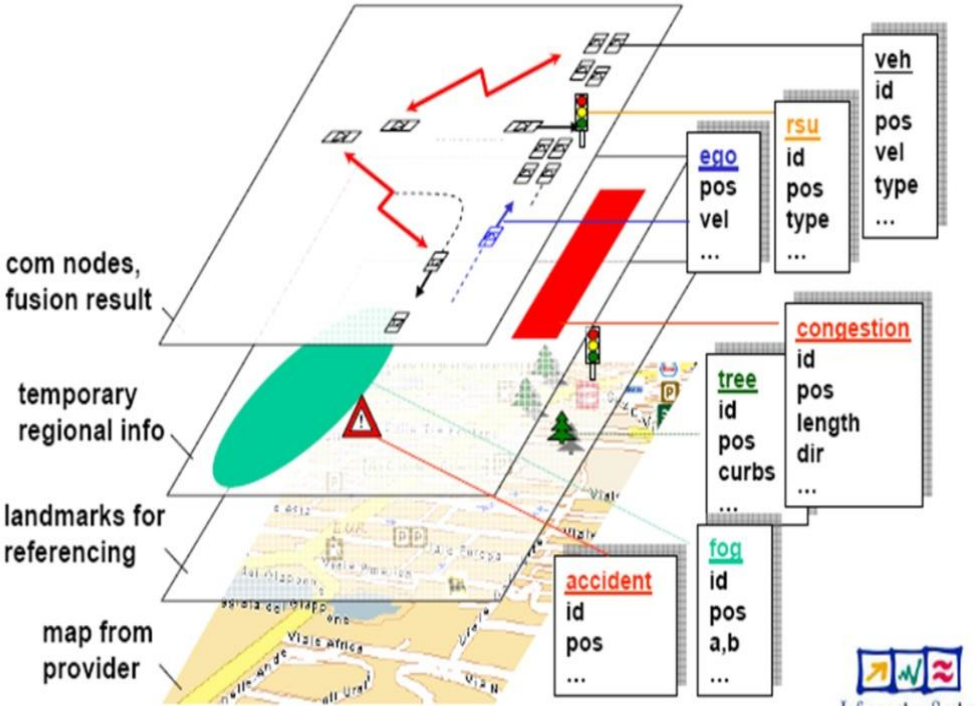


FIGURE 4.1.: *Carte locale dynamique (Projet Safespot [saf])*

- Le premier niveau contient les informations issues de la carte numérique. Cette carte, qui décrit la géométrie et la connexité des routes, est directement accessible à l'aide d'un système d'information géographique (SIG). Chaque route est décomposée en tronçons (entre chaque intersection) et chaque tronçon est décrit par une suite de segments.
- Le deuxième niveau contient les amers visuels que peut utiliser un système de perception embarqué pour se repérer et améliorer sa localisation par rapport à la carte. Les amers visuels sont généralement des objets statiques mais qui ne sont pas cartographiés dans les SIG classiques. Leur nombre et leur nature diffèrent selon les méthodes de localisation utilisées.
- Le troisième niveau décrit des phénomènes temporaires (brouillard, congestion, accident) qui peuvent concerner toute une zone de la LDM. La durée de vie de ces éléments est limitée et variable selon la nature de l'information.
- Enfin, le dernier niveau contient les éléments très mobiles de la scène à savoir le véhicule expérimental (ego-véhicule), les autres véhicules, les piétons, les 2 roues, etc. Ce niveau sert à prévenir des éventuelles collisions et permet

d'envisager la conduite coopérative.

Nous nous intéressons au dernier niveau de la carte locale dynamique. En particulier, nous nous plaçons dans le cadre de la *localisation* et de la *perception coopératives*. La *localisation coopérative* permet d'améliorer la localisation de chaque véhicule. Dans cette situation, le véhicule se localise et estime la localisation des autres véhicules dans son environnement. La *perception coopérative* permet d'augmenter le champ de perception de chaque véhicule en coopérant avec les autres. Dans les deux situations, le véhicule utilise ses capteurs pour se localiser et détecter son environnement et il communique avec ses voisins pour échanger des données. Les données échangées diffèrent selon l'application envisagée.

La localisation coopérative a été traitée dans différents domaines tels que la robotique coopérative et les applications de véhicules intelligents. Les approches de localisation coopérative varient selon les données envoyées aux autres membres du groupe. On peut distinguer deux classes d'approches : chaque agent peut envoyer ses propres données fournies par ses propres capteurs (approche coopérative : partie gauche de la figure 4.2) ou envoyer les données reçues par les autres (approche distribuée : partie droite de la figure 4.2). Un état de l'art sur la localisation coopérative est fait dans [CMHC13].

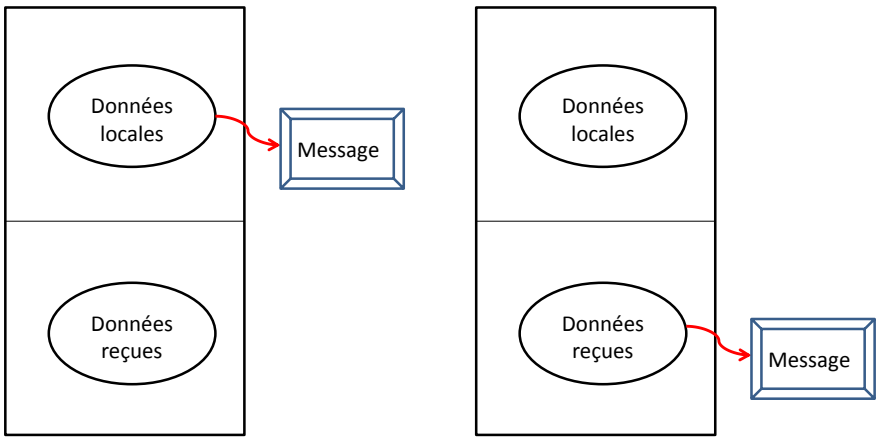


FIGURE 4.2.: *Les données envoyées par un véhicule sous forme de message pendant l'application de la localisation coopérative*

Message contenant des données locales

Différents travaux ont évoqué la localisation coopérative en envoyant les données locales détectées par les propres capteurs du véhicule. Roumeliotis et al. [RB02] considèrent un système centralisé composé de robots individuels capables de se déplacer, de détecter et communiquer avec les autres robots. Chaque robot est équipé de capteurs proprioceptifs pour mesurer son déplacement et de capteurs extéroceptifs pour observer l'environnement, détecter les robots mobiles aux alentours et mesurer leurs déplacements relatifs. Chaque robot échange sa position et son orientation avec l'ensemble du groupe. Le système de fusion utilisé est le filtre de Kalman. Les auteurs démontrent qu'on peut passer d'un système de fusion par filtrage de Kalman centralisé à un système décentralisé. Le filtre de Kalman centralisé combine les informations collectées par le groupe de robots et donne une estimation de la position de chaque membre du groupe. Le filtre de Kalman décentralisé est local pour chaque robot, il traite les données détectées par son propre robot. A chaque fois les robots se rencontrent, ils mesurent leurs poses relatives et leurs orientations et échangent leurs estimations pour les mettre à jour. Cette méthode a été validée sur un exemple de trois robots. Cependant, elle manque de généralité car les auteurs n'ont pas considéré les cas où les capteurs ne fournissent pas les données de pose relative, c'est-à-dire la position et l'orientation.

Martinelli et al. [MPS05] se sont basés sur les travaux de [RB02] et ont introduit une approche basée sur le filtre de Kalman étendu EKF en considérant l'observation relative la plus générale entre deux robots. Ils ont considéré trois observations différentes : la distance relative, l'orientation relative et la direction relative. Leur méthode a été validée par simulation.

Karam et al. dans [KCAC06] estiment l'état du groupe. Chaque véhicule possède deux estimations de son état de groupe. La première estimation est mise à jour en utilisant les données capteurs de l'égo-véhicule. Cette information est envoyée aux autres véhicules. Elle n'est pas mise à jour avec les données reçues des autres véhicules. La deuxième estimation est mise à jour avec les données reçues ; elle est conservée par le véhicule lui même et n'est pas envoyée aux autres. Leur méthode a été validée par simulation sur un groupe de quatre véhicules.

Tischler & Hummel [TH05] proposent l'échange de deux types d'informations : l'état de chaque véhicule et celui des objets détectés. Les données reçues sont fusionnées avec les données du véhicule issues d'un radar. Les objets perçus par le

radar sont pistées par un filtre de Kalman et les mesures entre deux instants sont associées en utilisant la méthode cJPDA (cheap Joint Probabilistic Data Association) [BS00]. Les données reçues passent par une étape de transformation de repère et sont fusionnées avec les données du véhicule en utilisant un filtre de Kalman. Cette méthode est validée sur des données réelles.

Message contenant données publiques ou distribuées

La deuxième classe d'approches consiste à envoyer les données reçues par les autres. Ces données peuvent être envoyées telles quelles ou bien fusionnées avec les données locales. Dans ce dernier cas, il se peut que les mêmes données soient fusionnées plusieurs fois, d'où la nécessité d'un traitement approprié. Li & Nashashibi [LN12] ont présenté une localisation coopérative en utilisant le filtre à intersection de covariance. Leur méthode se base sur l'estimation de l'état de groupe. Celui-ci est décomposé en deux parties : la pose de l'égo-véhicule et les estimations de son voisinage local. Cette estimation est envoyée aux véhicules voisins. L'état du groupe est mis à jour avec les données capteurs de l'égo-véhicule et les estimations envoyées par les autres véhicules. Leur méthode est validée par simulation et permet de diminuer l'erreur de localisation des véhicules.

Les auteurs dans [CMHC13] présentent une approche de localisation coopérative basée sur le filtre CPI-EnKF (Common Past-Invariant Ensemble Kalman filter) qui est une mise à jour optimale du filtre de Kalman même en présence d'une corrélation entre l'estimation d'état et les erreurs d'observation. Les véhicules échangent l'estimation de position globale, le déplacement relatif et la covariance et utilisent les données reçues pour mettre à jour leur estimation de position locale. La validation de cette méthode est faite par simulation. Elle permet l'amélioration de l'estimation d'état des véhicules.

Notre approche pour la construction d'une carte dynamique distribuée consiste pour chaque véhicule à envoyer l'information qu'il a reçu des autres en considérant sa propre détection. Nous ne faisons pas une localisation coopérative puisque nous n'estimons pas les positions relatives. Nous nous intéressons plutôt à la perception coopérative où les véhicules coopèrent pour augmenter leur champ de perception. Les travaux traitant de la perception coopérative dans les domaines de la robotique et des véhicules intelligents sont assez rares. Le projet URUS (Ubiquitous networking Robotics in Urban Settings) [uru] avait comme tâche de développer et

implémenter un outil de perception coopérative entre robots hétérogènes. Le projet Ko-PER [kop] utilise un réseau de capteur distribué pour réaliser la perception coopérative. Les informations sur l'environnement sont détectées par les capteurs des véhicules et renforcées par des réseaux de capteurs fixes dans des endroits comme les intersections. Le projet CooPerCom (Cooperative Perception and Communication in vehicular technologies) [coo] a pour objectif de développer des approches coopératives et distribuées et des outils de traitement de l'information pour mettre à profit la communication inter-véhicules afin de récupérer les données, les traiter et évaluer leurs incertitudes. Li et Nashashibi [LN11] ont présenté une méthode de perception coopérative pour une application de réalité augmentée. Leur méthode a été appliquée sur un exemple de deux véhicules. L'idée est de transformer la partie occultée du véhicule avant en une perception visuelle du véhicule arrière en se basant sur la perspective 3D. Tout d'abord, les auteurs estiment la pose relative entre deux véhicules de référence afin d'avoir un repère commun pour les perceptions des véhicules, ce qui leur permet de transformer les perceptions des véhicules en perspective 3D. Leur méthode a été testée sur des véhicules expérimentaux.

Nous considérons que la carte locale dynamique d'un véhicule est constituée des objets mobiles détectés dans la scène et nous souhaitons échanger ces cartes entre les véhicules afin d'augmenter le champ de perception de chacun d'eux (figure 4.3). Chaque véhicule va envoyer sa pose ainsi que les objets détectés. Le véhicule va garder pour lui ce qu'il a détecté par ses propres capteurs, et il va envoyer la mise à jour de ce qu'il a détecté avec ce qu'il a reçu. Nous nous basons sur l'algorithme de fusion distribuée introduit dans le chapitre 2 afin d'établir une confiance sur l'existence des objets dans une scène. Pour simplifier les notations, dans ce qui suit, nous remplaçons la notion de carte locale dynamique par celle de carte dynamique CD et nous introduisons les notions de carte dynamique locale CDL et de carte dynamique distribuée CDD. Cette différence vient du fait que le véhicule aura deux bases de connaissances : locale et distribuée. Il faut noter que la connaissance distribuée dans ce chapitre fait référence à la connaissance publique mentionnée dans les chapitres précédents.

Dans ce chapitre, nous présentons le problème de la construction d'une carte dynamique distribuée. Nous décrivons comment l'algorithme de fusion distribuée a été utilisé dans ce contexte ainsi que les résultats obtenus.

FIGURE 4.3.: *Illustration de la perception coopérative où les véhicules échangent des informations pour augmenter leur champ de perception (www.car-to-car.org)*

4.2. Formalisation du problème de la carte dynamique distribuée

4.2.1. Carte dynamique et pose

Nous considérons que chaque véhicule V_j détecte par ses capteurs une carte dynamique notée CD_j qui n'est autre qu'une liste d'objets O_i (où i représente l'identifiant de l'objet). Muni d'un système de localisation absolue et d'une horloge commune, le véhicule V_j connaît son état représenté par $X_{V_j} = (x_j, y_j, \varphi_j, v_j, P_j)$ où x_j, y_j représentent la position du véhicule, φ_j le cap, v_j la vitesse du véhicule et P_j représente les erreurs associées. La figure 4.4 illustre une situation où deux véhicules sont munis de capteurs et détectent des objets dans la scène. Le véhicule E détecte trois objets : e_1, e_2, e_3. Il n'arrive pas à détecter l'objet f_1 puisqu'il est occulté par l'objet e_1. F détecte aussi trois objets qui sont dans son champ de vision (f_1, f_2, f_3).

La carte dynamique CD_j est constituée d'une liste d'objets $\{O_i\}_j$; chaque objet a les attributs suivants dans le repère associé à V_j :

- p_i la position de l'objet i,
- v_i la vitesse de l'objet i,
- c_i la classe à laquelle appartient l'objet i, cette information dépend du type du capteur utilisé pour détecter la scène,
- σ_{p_i},σ_{v_i} les incertitudes sur la position et sur la vitesse ou la matrice de covariance P_i,
- m_{c_i} est la masse d'appartenance à la classe, dont le calcul sera détaillé dans la section 4.5,
- m_i masse modélisant l'existence de objets $\Omega = \{O, \overline{O}\}$

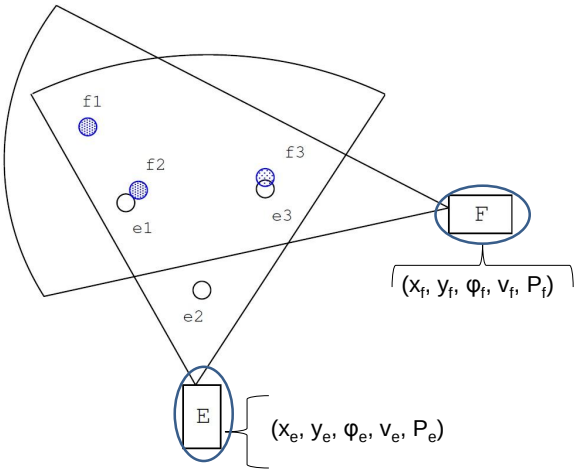


FIGURE 4.4.: *Exemple de situation : Véhicules E et F sont munis de capteurs et détectent les objets dans la scène*

t_j	Id_j	Etat Xv_j	c_j	$CD_j=\{O_i, m(O_i)\}_j$

FIGURE 4.5.: *Carte dynamique du véhicule V_j où t_j est le temps de la pose, Id_j l'identifiant du véhicule, X_{V_j} son état, c_j sa classe et sa CD_j.*

Ces informations dépendent du type du capteur utilisé par les véhicules. La figure 4.5 montre la carte dynamique du véhicule V_j. Par exemple, le véhicule F de la figure 4.4 à l'instant t_f a comme état $(x_f, y_f, \varphi_f, v_f, P_f)$ et il appartient à la classe véhicule et sa carte dynamique est $CD_f = \{f_1, m(f_1), f_2, m(f_2), f_3, m(f_3)\}$.

Le véhicule V_j donne un avis sur l'existence de l'objet détecté, représenté par $m_j(O_i)$. Nous cherchons à évaluer la confiance sur l'existence des objets détectés dans une scène. Nous nous intéressons à l'existence de l'objet dans le but d'augmenter le champ de perception de chaque véhicule et de diminuer les fausses alarmes positives. Le cadre de discernement sera alors : $\Omega = \{O, \overline{O}\}$ où "O" représente l'existence des objets (voiture, camion, bus, etc.) et "$\overline{O}$" les non-objets (fausses alarmes dues à la détection de la route, du trottoir, etc.). Les fonctions de masses sur l'exis-

tence de l'objet sont construites de la manière suivante :

$$\begin{array}{lcl} m(O) & = & a_{ij}, \\ m(\overline{O}) & = & b_{ij}, \\ m(\Omega) & = & 1\text{-}a_{ij}\text{-}b_{ij}. \end{array} \tag{4.1}$$

Un exemple d'attribution de ces masses est donné dans la section 4.5.1.

4.2.2. Connaissances locale et distribuée

Comme mentionné dans le chapitre 2, chaque véhicule (noeud) V_j possède deux types de connaissances :

– Connaissance locale : $CDL_j(t)$,
– Connaissance distribuée : $CDD_j(t)$.

La connaissance locale CDL est ce que le véhicule V_j détecte avec ses propres capteurs ; elle contient la masse modélisant la croyance de l'existence réelle qu'il attribue aux objets détectés (figure 4.6). Le véhicule garde cette connaissance pour lui, il ne l'envoie pas directement aux autres véhicules mais la combine avec les messages reçus des autres pour établir la connaissance distribuée CDD. Le résultat appelé CDP (carte dynamique publique) sera envoyé aux autres véhicules (figure 4.7). Les connaissances seront représentées par des vecteurs de masse puisqu'il s'agit de différents éléments d'observation.

Afin d'exploiter les messages, chaque véhicule doit transformer les données reçues dans un repère global qui est le repère Monde, recaler temporellement les données et associer les différents objets détectés en utilisant l'algorithme d'association optimale détaillé dans le chapitre 5. Après toutes ces étapes, il pourra mettre à jour sa connaissance distribuée.

4.3. Carte dynamique locale : CDL

La connaissance locale est établie à partir d'une connaissance directe (algorithme 3 du chapitre 2). La connaissance directe dans cette application provient des données issues des capteurs qui détectent l'environnement dynamique du véhicule. La carte dynamique locale CDL peut être construite de trois façons différentes, selon le type et le nombre de capteurs utilisés dans l'application. Ces trois cas sont

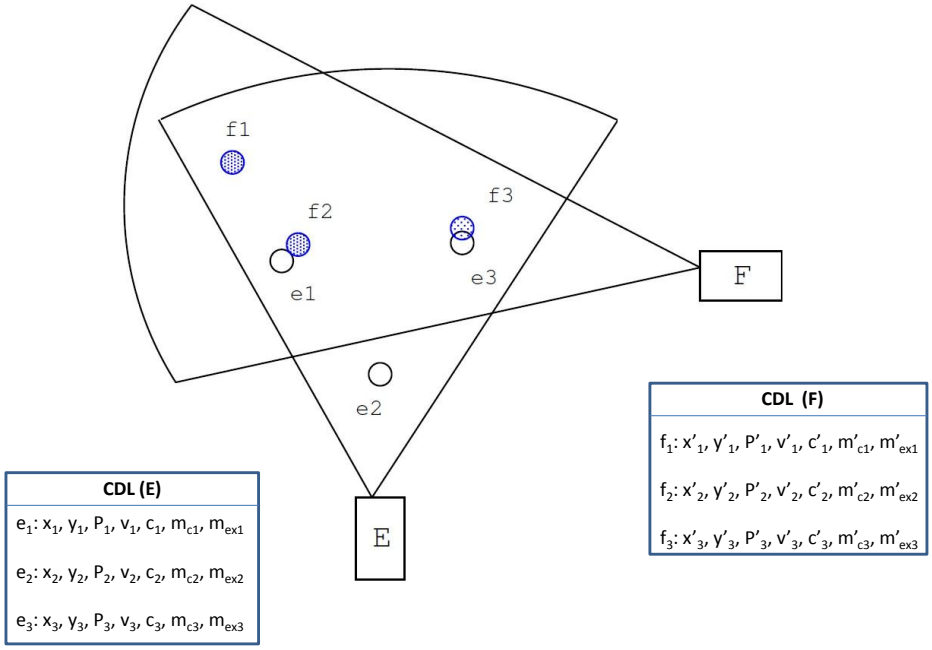


FIGURE 4.6.: *Exemple de carte dynamique locale*

résumés dans ce qui suit :

1. Le véhicule reçoit des informations de plusieurs capteurs. Dans ce cas, il faut fusionner les données provenant des sources pour construire sa carte.

2. Le capteur fournit des observations qui nécessitent un algorithme de pistage. Il faut suivre les objets, c'est-à-dire fusionner les cartes entre les instants t et $t-1$ afin d'éliminer les fausses alarmes.

3. Le véhicule est équipé d'un capteur intelligent qui fournit des pistes. La carte locale peut alors être définie directement à partir des pistes.

Dans notre cas, le véhicule est muni d'un capteur intelligent qui fournit des pistes (un pisteur). Nous construisons directement ce qu'on appelle la carte dynamique locale CDL à partir des données capteurs. Ce traitement sera détaillé dans la section 4.5. Nous proposons une solution pour les deux autres cas de construction de la CDL dans l'annexe B.

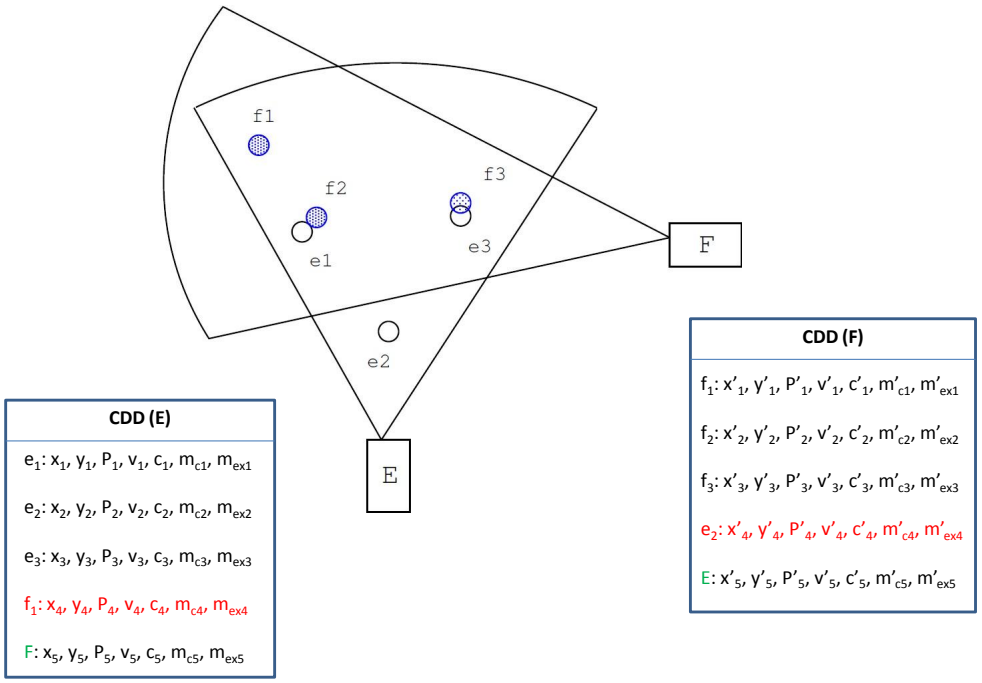


FIGURE 4.7.: *Exemple de carte dynamique distribuée si E et F échangent leurs cartes*

4.4. Construction de la carte dynamique distribuée

L'algorithme 5 présente une version de l'algorithme 3 pour la fusion distribuée adaptée à la construction de la carte dynamique distribuée CDD. Ce traitement se fait du côté du récepteur lorsqu'il reçoit un message contenant la carte de l'émetteur notée par un indice e. La carte du récepteur est indicée par r.

La figure 4.8 montre le principe de l'algorithme de construction de la carte dynamique distribuée. La CDL est construite directement à partir des données issues des capteurs. La CDD est mise à jour à partir des messages en utilisant la règle prudente ($\oslash$). Les CDL et CDD sont combinées dans CDP en utilisant la règle de Dempster ($\oplus$).

Algorithme 5 : Fusion distribuée à la réception d'un message : cas de la carte dynamique distribuée

Données : Message de l'émetteur CDP_e ;CDD_r ;
Résultats :CDP_r ;CDD_r ; ;
Injection du message
$CDP_e =^{\alpha} CDP_e + E$: Affaiblissement de la CDD_e de l'émetteur et ajout de l'émetteur dans sa liste d'objet;
$\widehat{CDP_e} = prediction(CDP_e)$;
$\widehat{CDD_r} = prediction(CDD_r)$;
$CDD_r \leftarrow FusionCautious(\widehat{CDP_e}, \widehat{CDD_r})$;
Injection de la connaissance locale CDL_r
$CDL_r \leftarrow acquisition\ connaissance\ locale()$;
$CDP_r \leftarrow FusionDempster(CDD_r, CDL_r)$;
Envoi CDP_r;

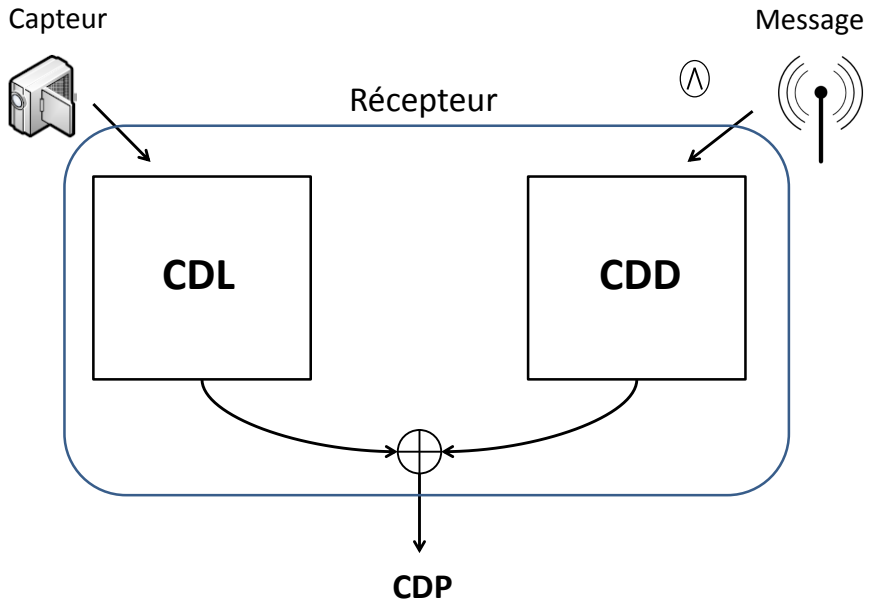


FIGURE *4.8.: Principe de l'algorithme de CDD*

4.4.1. Mise à jour de la connaissance distribuée

Injection des messages

Comme mentionné dans le chapitre 2, les masses dans la connaissance distribuée (publique) reçue doivent être affaiblies pour garantir la convergence de l'algorithme en cas d'erreur. A la réception du message contenant CDP_e, le récepteur affaiblit la carte de l'émetteur ${}^{\alpha}CDP_e$ et ajoute l'émetteur E à sa carte. L'affaiblissement se fait sur les masses relatives à l'existence des objets de la façon suivante :

$$ {}^{\alpha}m = (1-\alpha).m(A) + \alpha.m(\Omega) \;\; avec\; A = \{O_e, \overline{O_e}\}. \tag{4.2} $$

Prédiction :
Afin de mettre à jour la carte distribuée du récepteur avec la carte contenue dans le message reçu, il faut faire un recalage temporel. Ceci consiste à prédire l'état à l'instant t du traitement avec un modèle à vitesse constante. D'une manière générale, on notera $\widehat{x}$ la prédiction de x. La fonction de prédiction ($\widehat{CDP_e}$= prediction(CDP_e), $\widehat{CDD_r}$= prediction(CDD_r)) permet de prédire les cartes de la façon suivante :

$$ O \begin{cases} \hat{x}(t') &= x(t) + v_x(t) * \Delta t, \\ \hat{y}(t') &= y(t) + v_y(t) * \Delta t, \\ P(t') &= F.P(t).F^T + Q, \\ m_c(t') &\leftarrow m_c(t), \\ v(t') &= v(t). \end{cases} \quad m(O) \Big\{ \; m(t') \leftarrow {}^{\alpha'}m(t). \tag{4.3} $$

où $\Delta t = t' - t$, t' est le temps de prédiction et t est le temps de construction de la carte, F est la matrice qui relie $P(t)$ à l'instant (t') avec celle à l'instant (t) et Q est la matrice de covariance. Les masses des cartes prédites subissent aussi un affaiblissement calculé en fonction du temps de la même façon que dans l'équation 2.7. Cet affaiblissement est représenté par α'.

Les cartes prédites sont ensuite associées en utilisant l'algorithme d'association du chapitre 5 qui utilise les informations sur la position, la vitesse et la classe pour trouver la relation entre les objets.

L'algorithme 6 prend en compte trois cas pour la fusion des objets :

– Si les objets (O_e, O_r) sont associés , nous mettons à jour la masse sur l'existence

Algorithme 6 : FusionCautious

Données : $\widehat{CDP_e}, \widehat{CDD_r}$;
Résultats : CDD_r;
$Association(\widehat{CDP_e}, \widehat{CDD_r})$;
  Pour chaque objet O_e et O_r associé
   $m_r^{(t)} \leftarrow m_r^{(t)} \textcircled{\wedge} m_e^{(t)}$
  Pour chaque O_r *seul*
   $O_r \leftarrow^{\alpha} O_r$
  Pour chaque O_e *seul*
   $CDD_r \leftarrow CDD_r + O_e$

$m_r^{(t)}$ par la règle prudente,

$$m_r^{(t)} = m_r^{(t)} \textcircled{\wedge} m_e^{(t)}. \tag{4.4}$$

– Si l'émetteur ne détecte pas un objet de la carte locale du récepteur (c'est-à-dire l'objet O_r est seul), la masse relative à l'existence de l'objet de la carte est affaiblie comme dans l'équation 4.2 ;
– Si l'émetteur fait état d'un objet inconnu par le récepteur alors on l'ajoute dans la CDD_r.

Injection de la connaissance locale

Après avoir extrait la carte dynamique locale CDL_r des informations données par le ou les capteurs du véhicule, on la combine avec la CDD_r. Pour cela, il faut tout d'abord associer les objets avant la mise à jour. Le résultat obtenu est stocké dans la carte dynamique publique CDP_r qui sera envoyée aux autres véhicules.

L'algorithme 7 montre les trois cas à prendre en compte pour la fusion des objets :

– Si les objets (O_r, O_{r_l}) sont associés, O_r est mis à jour par O_{r_l}. Les deux connaissances sont fusionnées par la règle de Dempster :

$$m_r^{(t)} = m_{r_l}^{(t)} \oplus m_r^{(t)}; \tag{4.5}$$

– Si l'objet de la connaissance locale O_{r_l} est seul, on le garde dans la CDP_r ;
– Si l'objet O_r de la connaissance distribuée CDD_r n'est pas associé et devrait être vu par le récepteur, on ne le garde pas. S'il n'est pas dans la zone de visibilité, on l'ajoute à CDP_r.

Algorithme 7 : FusionDempster

Données : CDL_r, CDD_r ;
Résultats :CDP_r ;
$CDP_r \leftarrow CDD_r$;
$Association(CDD_r, CDL_r)$;
  Pour chaque objet O_r et O_{r_l} associé
    $O_r \leftarrow O_{r_l}$
    $m_r^{(t)} = m_{r_l}^{(t)} \oplus m_r^{(t)}$
  Pour chaque O_{r_l} *seul*
    $CDP_r \leftarrow CDP_r + O_r$
  Pour chaque O_r *seul*
    if $Visible$ **then**
      $delete(O_r)$
else
    keep(CDP_r)

Finalement, la CDP_r est envoyé à travers le réseau.

Il faut noter que la connaissance locale dans l'algorithme de la carte dynamique distribuée n'est injectée qu'une fois, à cause du pistage qui intègre déjà dans le temps des données locales.

4.5. Implémentation et résultats

L'application de la carte dynamique distribuée a été validée en utilisant le simulateur développé en Matlab par A. Houenou [HBCB12]. Nous avons simulé deux scénarios impliquant différents véhicules parmi lesquels certains sont munis d'une caméra intelligente ayant comme : *champ de vision* $=45°$ et *distance*$= 60m$, d'un système de localisation de type GPS et d'une antenne wifi. Les véhicules munis d'une antenne wifi communiquent et échangent leurs cartes dynamiques distribuées. Ils circulent sur une route à plusieurs voies et détectent leur environnement.

Le capteur est un pisteur qui fournit différentes types d'informations :
- l'identifiant de l'objet,
- sa position relative x et y par rapport au véhicule et la matrice de covariance correspondante P_x,
- sa vitesse relative v_x et v_x par rapport au véhicule et la matrice de covariance correspondante P_v,

– l'âge de l'objet,
– la classe de l'objet.

A partir de toutes ces informations fournies par le capteur, nous construisons les masses sur la classe et l'existence de l'objet, ainsi que les cartes dynamiques.

4.5.1. Construction des masses pour la connaissance directe

Le capteur détecte les objets chaque 0.1s. L'âge de l'objet fourni par le capteur représente le nombre de fois où le pisteur a détecté cet objet. Nous construisons les masses de l'existence sur l'objet à partir de son âge de la manière suivante :

$$\begin{array}{rcl} m(O) & = & \alpha'.(1 - \mathrm{e}^{-k.age(O_j)}) \\ m(\overline{O}) & = & \alpha'.(\mathrm{e}^{-k.age(O_j)}) \\ m(\Omega) & = & (1 - \alpha') \end{array} \quad (4.6)$$

où $\alpha' = 0.9$ représente la fiabilité du capteur et $k = 0.1$. La figure 4.9 montre la masse sur l'existence en fonction de l'âge. Plus on a détecté l'objet, plus la masse sur l'existence est renforcée.

En ce qui concerne la masse sur la classe, le simulateur ne fournit pas cette information. Le cadre de discernement de la classe est $\Omega = \{V, \overline{V}\}$ où V représente le fait que l'objet soit un véhicule et $\overline{V}$ ne soit pas un véhicule. Nous attribuons une masse 0.9 à $\{V\}$ ou $\{\overline{V}\}$ selon la décision du capteur (l'objet est un véhicule ou non) et une masse 0.1 sur $\Omega = \{V, \overline{V}\}$.

4.5.2. Recalages spatial et temporel

Recalage spatial : Chaque véhicule détecte les objets dans son propre repère noté M et construit sa carte dynamique locale. Les repères du capteur et du véhicule étant les mêmes dans le simulateur, on n'a pas besoin de faire un changement de repère entre capteur et véhicule. La transformation de repère se fait à deux niveaux :

– A la réception d'un message, le véhicule récepteur transforme l'état des objets dans le repère absolu commun à tous les véhicules, appelé repère Monde et noté W (pour world).
– Au moment de l'injection de la connaissance locale dans la connaissance dis-

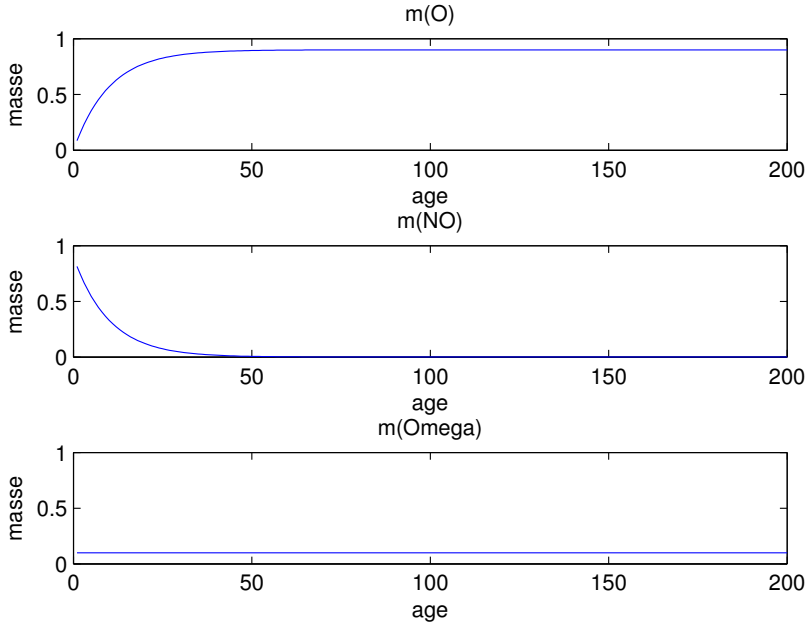


FIGURE *4.9.: La masse sur l'existence en fonction de l'âge*

tribuée, il faut exprimer la connaissance locale dans le repère monde avant de les combiner.

La figure 4.10 représente les deux repères M (associé au véhicule), W et l'objet O. Soient (x', y') la position de l'objet O dans le repère véhicule M et (x, y) sa position dans le repère monde W et soit $(\Delta x, \Delta y)$ la pose du repère M dans W. Le changement de repère est fait de la façon suivante :

$$\begin{aligned} x &= x'.cos(\varphi) + y'.sin(\varphi) + \Delta x. \\ y &= -x'.cos(\varphi) + y'.sin(\varphi) + \Delta y. \end{aligned} \tag{4.7}$$

Recalage temporel : Dans une telle application, il faut recaler les données issues des capteurs et les messages. Nous considérons dans cette simulation que les capteurs ont une horloge commune. Chaque véhicule détecte les objets, remplit sa CDL, reçoit des messages, met à jour sa CDD, la combine avec sa CDL et envoie sa CDP.

Le capteur détecte les objets avec un pas de temps $T = 0.1s$. Pour simplifier, nous supposons que le véhicule envoie les messages des données traitées à la même période. Le temps de transmission de message n'est pas maîtrisé mais il est borné.

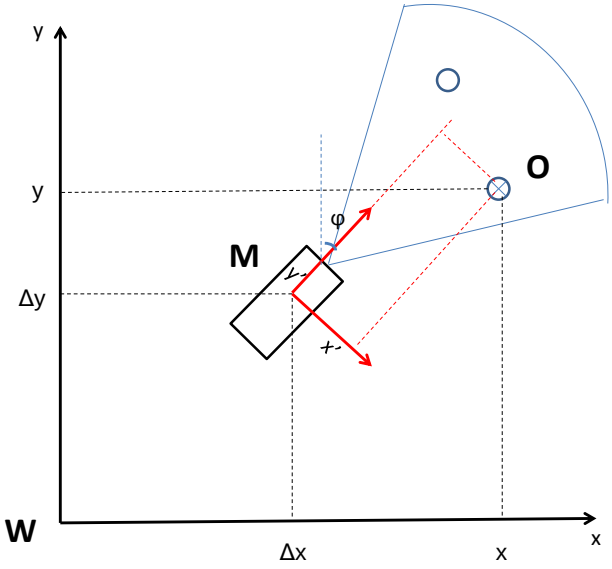


FIGURE *4.10.: Changement de repère entre le repère véhicule M et le repère monde W*

Nous avons alors choisi de prédire les données et de les fusionner à l'instant présent comme le montre la figure 4.11 avant d'envoyer les messages. Quelque soit le temps d'arrivée des messages, toutes les données sont synchronisées et traitées au même moment. Comme le montre l'algorithme 5, chaque véhicule à la réception d'un message va prédire, en utilisant la fonction $prediction()$ à l'instant de réception, sa CDD_r à l'instant précédent ainsi que le message reçu et fusionne ces deux données à l'instant (t).

4.5.3. Scénarios

Scénario 1 : Trois véhicules équipés et un non équipé

Nous avons créé un scénario avec quatre véhicules $\{V_0, V_1, V_2, V_3\}$. Les véhicules $\{V_0, V_1, V_2\}$ sont munis d'une caméra intelligente et d'une antenne wifi. Tous les véhicules équipés peuvent recevoir le message envoyé. V_0 et V_1 se suivent, V_3 les dépasse et V_2 roule en sens inverse. La figure 4.12 montre le scénario à différents pas de temps.

Les capteurs détectent les objets chaque $0.1s$. Chaque véhicule construit sa CDL,

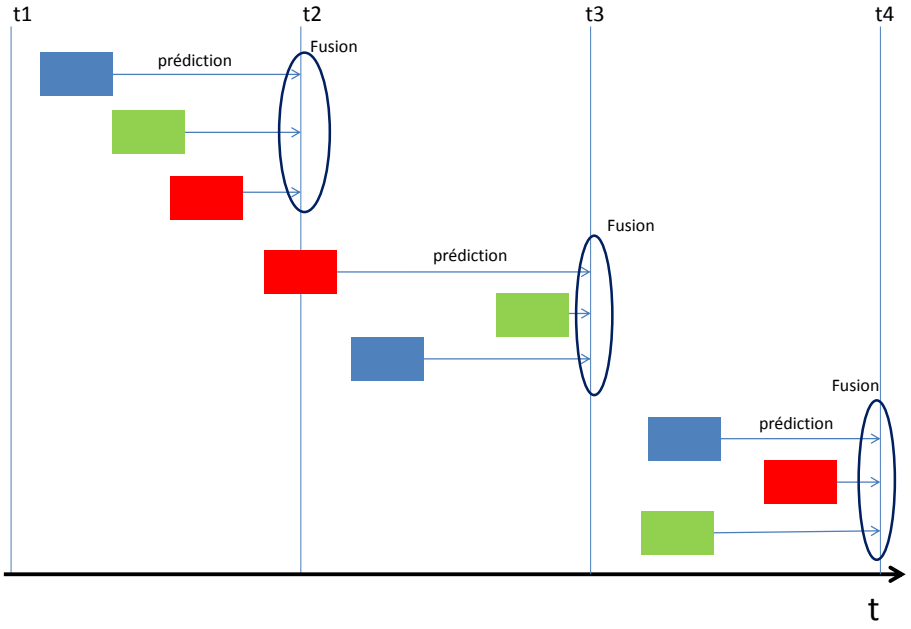


FIGURE 4.11.: *Recalage temporel : Prédiction avant traitement des données. Les couleurs bleu, vert et rouge représentent les données de 3 véhicules différents.*

met à jour sa CDD avec les données disponibles (messages reçus) et envoie les messages chaque $0.1s$. Le scénario se déroule pendant 15s. La figure 4.13 montre le résultat du scénario 1 à l'instant 2.7s. La première colonne représente la vérité terrain (VT), la seconde colonne montre la CDL et la troisième colonne la CDP (carte distribuée envoyée). Le capteur de chaque véhicule détectant la scène est dessiné en rouge.

CDL : V_0 détecte V_1 et V_3, cette détection est représentée par les triangles noirs dans la CDL V_0. Les véhicules V_1 et V_2 ne détectent rien. Les véhicules $\{V_0, V_1, V_2\}$ communiquent entre eux.

CDP : A chaque réception de message, le véhicule récepteur ajoute le véhicule émetteur à sa liste. La CDP de chaque véhicule est représentée par des carrés noirs. Les véhicules V_1 et V_2 ne détectent pas les autres véhicules mais les ajoutent en recevant leurs messages. Par exemple, V_1 ajoute V_0 et V_2 à sa CDD en recevant un message d'eux et ajoutent V_3 à sa liste puisqu'il a été détecté par V_0. Les véhicules de la vérité terrain sont représentés sous forme de points colorés dans la CDP pour simplifier la vérification. En échangeant des messages, les véhicules augmentent

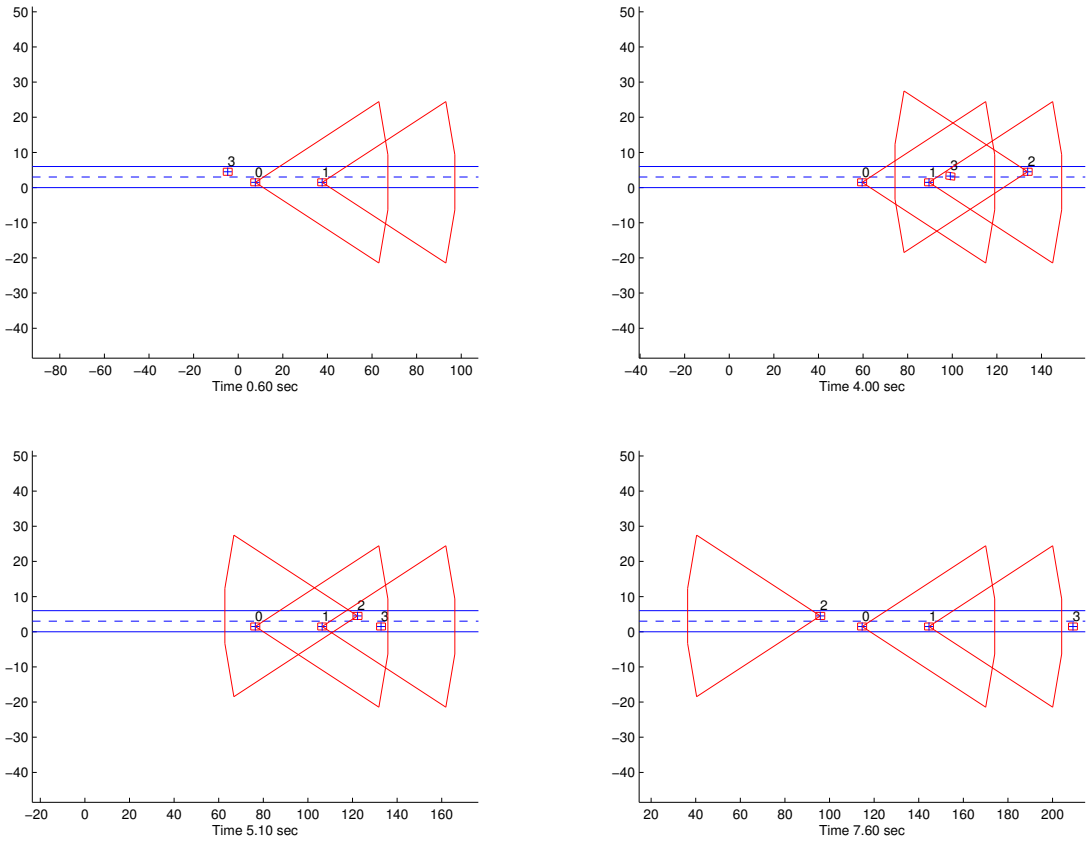


FIGURE 4.12.: *Scénario 1 à différents pas de temps. $\{V_0, V_1, V_2, V_3\}$ sont des véhicules équipés. V_0 et V_1 se suivent, V_3 les dépasse et V_2 roule en sens inverse*

leur champ de vision.

Scénario 2 : Cinq véhicules équipés et trois non équipés

Le scénario 2 inclut sept véhicules $\{V_0, V_1, V_2, V_3, V_4, V_5, V_6, V_7\}$ qui circulent pendant $25s$. Les véhicules $\{V_0$,V_2,V_3,V_4,$V_5\}$ sont équipés d'un capteur et d'une antenne wifi alors que $\{V_1, V_6, V_7\}$ sont non équipés. Le scénario se déroule de la façon suivante : V_0, V_1 et V_2 se suivent (figure 4.14a). V_1 à partir de l'instant $4.3s$ dépasse V_2 (figure 4.14b). V_5 roule dans le sens inverse et détecte V_1 et V_2, puis V_0 (figure 4.14b). V_0, V_1 et V_2 arrivent à une intersection et ralentissent, V_3 et V_4 circulent dans le sens perpendiculaire et se croisent (figure 4.14c). V_6 arrive dans le même sens que V_3 et V_7 suit V_4 (figure 4.15a et 4.15b).

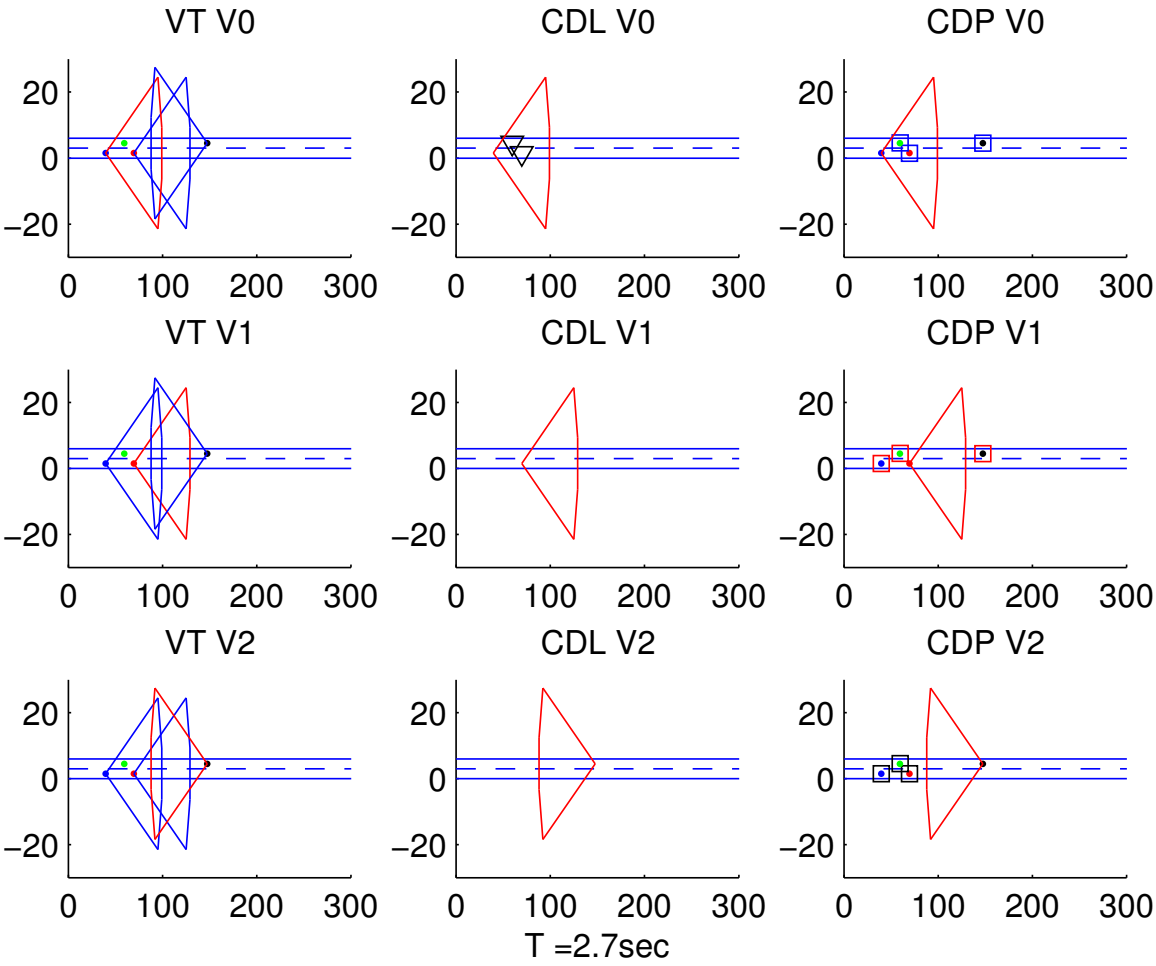
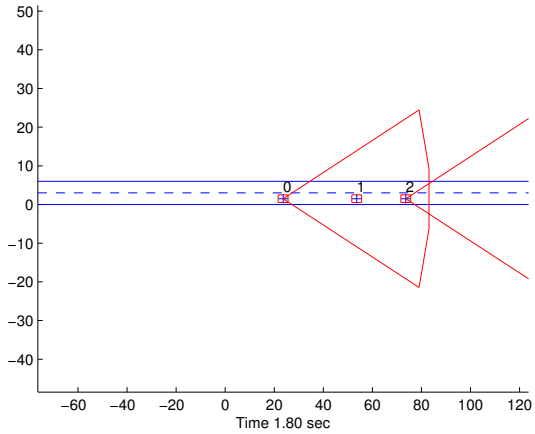


FIGURE *4.13.: Scénario 1 à l'instant 2.7s : la première colonne représente la vérité terrain (VT) de chaque véhicule, la seconde colonne montre la CDL et la troisième colonne la CDP.*

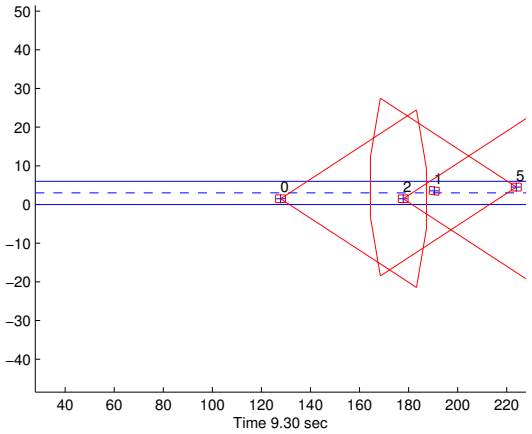
4.5.4. Résultats

Pour illustrer le résultat de la simulation, nous allons montrer les points de vues des véhicules équipés à différents instants dans les figures 4.16, 4.17 et 4.18. La figure 4.16a montre la vérité terrain, la CDL et la CDP de V_0. Celui-ci détecte dans sa CDL deux véhicules V_1 et V_2. Il reçoit des messages des véhicules de V_3, V_4, V_5, et les ajoute à sa CDP comme le montre la partie CDP V0.

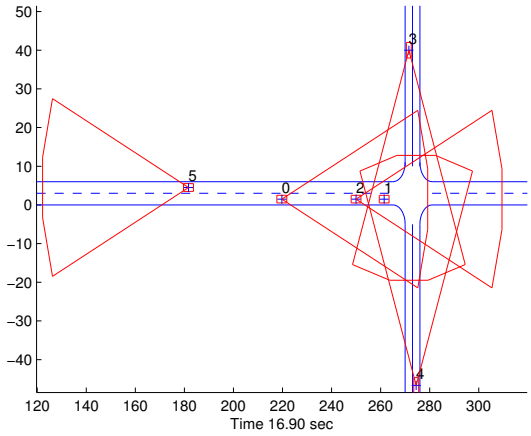
V_2 ne détecte pas V_1 ni les autres (figure 4.16b). Il ajoute V_1 à sa CDD puisqu'il a été détecté par V_0 ainsi que les trois autres véhicules V_3, V_4, V_5 puisqu'il a reçu leurs messages. V_3, V_4 et V_5 ne détectent rien mais, en échangeant des messages, ils



(a) V_0, V_1 et V_2 se suivent



(b) V_1 à partir de l'instant $4.3s$ dépasse V_2. V_5 roule dans le sens inverse et détecte V_1 et V_2, puis V_0

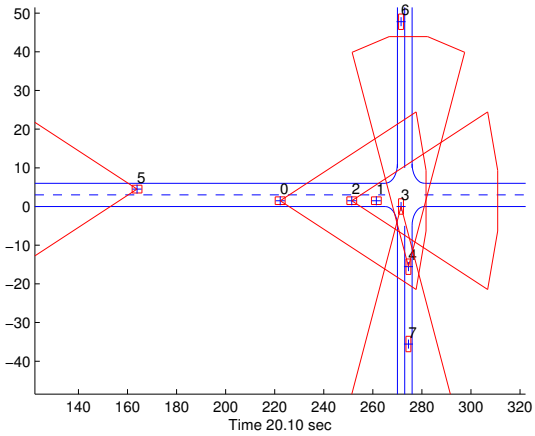


(c) V_0, V_1 et V_2 arrivent à une intersection et ralentissent, V_3 et V_4 circulent dans le sens perpendiculaire et se croisent

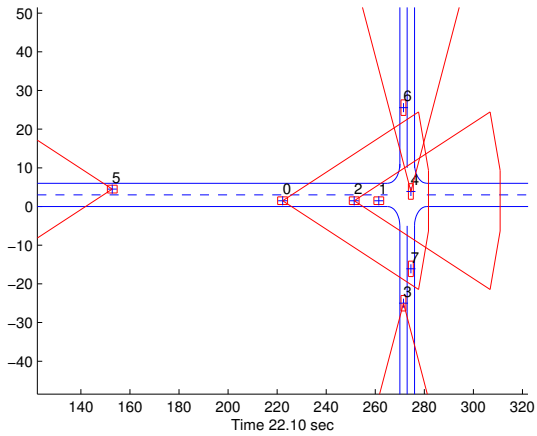
FIGURE 4.14.: *Scénario 2 à différents pas de temps.*

ont une connaissance globale à cet instant sur les véhicules équipés et les véhicules détectés (figures 4.17a, 4.17b, 4.18a). Nous remarquons que tous les véhicules ne connaissent pas V_6 ni V_7 puisque ces derniers ne sont pas équipés de capteurs et n'ont pas été détectés par un autre véhicule.

Les figures 4.19,4.20 et 4.21 montrent les connaissances des cinq véhicules à l'instant t=20s. Nous constatons que le véhicule V_7 a été détecté par le véhicule V_3 qui l'a envoyé aux autres véhicules. V_7 sera ajouté dans la carte des véhicules



(a) V_6 arrive dans le même sens que V_3



(b) V_7 suit V_4

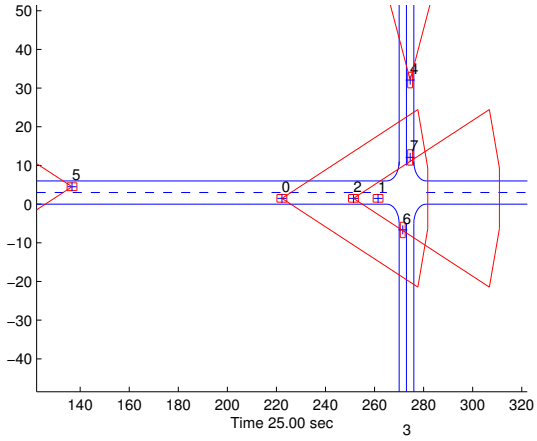


FIGURE *4.15.: Scénario 2 à différents pas de temps.*

recevant le message de V_3.

Ce scénario montre que même si les véhicules ne détectent pas les objets, ils les ajoutent et augmentent leur champ de vision en recevant des messages.

Afin d'évaluer l'application de la carte dynamique distribuée, nous avons effectué des tests sur le premier scénario. Ces tests portent sur l'évolution de la masse sur l'existence d'un objet non équipé. Nous donnons aussi les performances en comparant les taux de vrai positif (VP), de faux positif (FP) et de faux négatif (FN) des détections des trois véhicules V_0, V_1 et V_2 dans les cas distribué et local. Les résultats sont présentés dans ce qui suit.

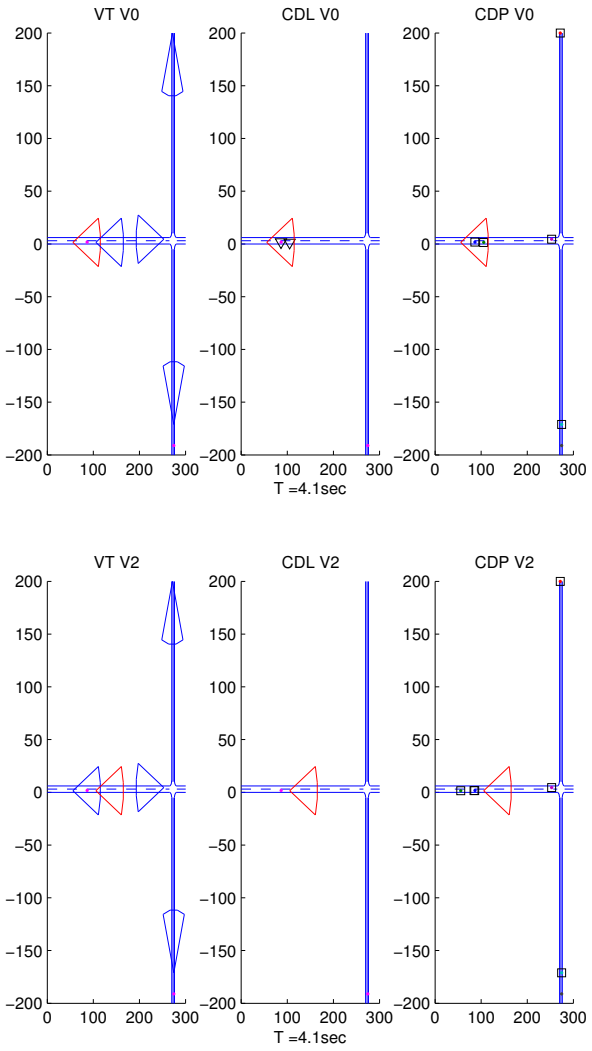


FIGURE *4.16.: Résultats du Scénario 2 à l'instant 4.1s : (a) VT, CDL et CDP du V_0, (b) VT, CDL et CDP du V_2.*

Evolution de la masse sur l'existence

Pour prendre la décision concernant l'existence de l'objet après la mise à jour de la carte distribuée, nous calculons la probabilité pignistique comme détaillé dans l'équation 1.19 de la façon suivante :

$$BetP(\overline{O}) = m(\overline{O}) + \frac{m(\Omega)}{2}. \tag{4.8}$$

On décide que cet objet n'existe pas si sa $BetP(\overline{O})$ est supérieure à un seuil ϵ prédéfini. L'objet est alors effacé de la carte distribuée.

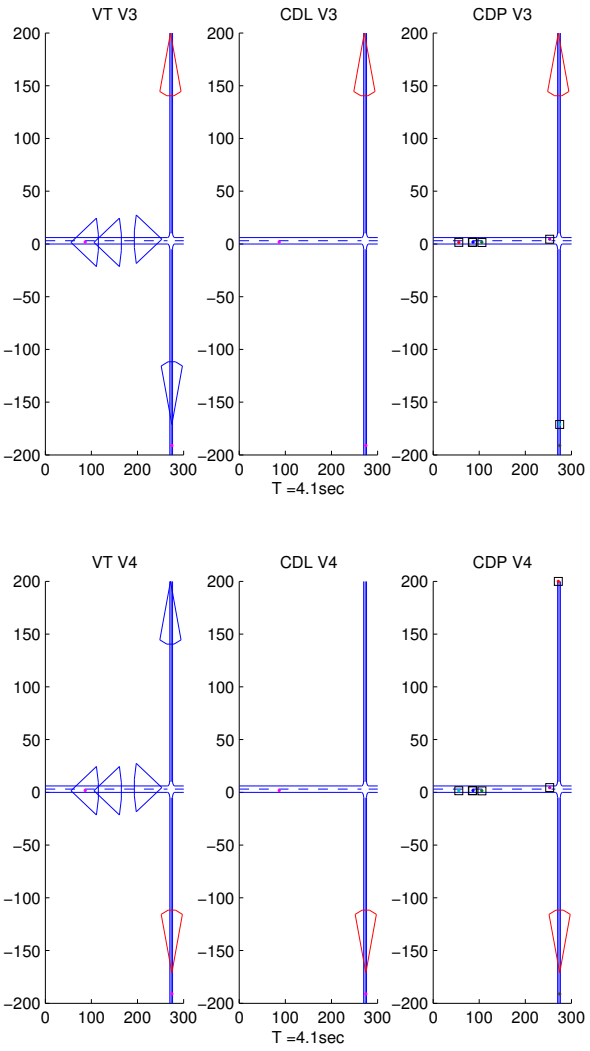


FIGURE 4.17.: *Résultats du Scénario 2 à l'instant 4.1s : (a) VT, CDL et CDP du V_3, (b) VT, CDL et CDP du V_4.*

La figure 4.22 montre l'évolution de la masse sur l'existence du véhicule 3 dans les cartes locale et distribuée des véhicules V_0, V_1 et V_2. Le véhicule V_3 n'est pas toujours détecté par tous les véhicules. Les courbes bleues représentent la masse relative à l'existence dans la carte locale et les rouges celles de la carte distribuée de chaque véhicule. La première colonne représente $m(V_3)$ et la deuxième colonne montre $m(NV_3)$ où NV_3 correspond à $\overline{V_3}$. V_0 détecte en premier V_3 à l'instant 2s. Il ne le garde pas dans sa carte distribuée avant que la probabilité pignistique atteigne un seuil ϵ. Pour cette raison, la masse sur l'existence de la carte distribuée de V_0 n'évolue pas avant l'instant 2.5s. Ensuite, V_0 envoie sa carte distribuée à V_1 et V_2. Cela explique l'apparition de V_3 dans la carte distribuée de V_1 entre 2.5 et 3.8s et

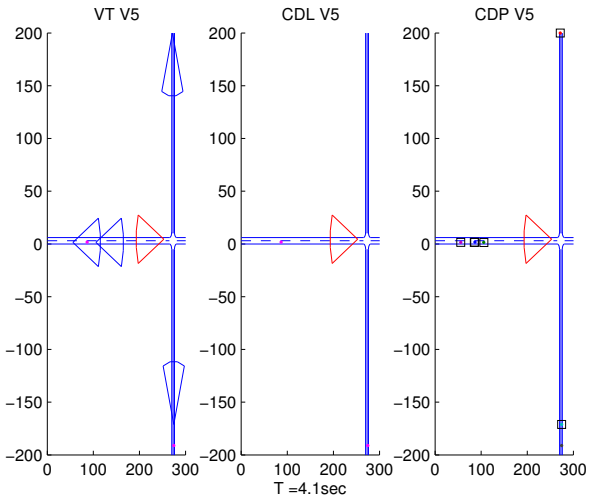


FIGURE *4.18.: Résultats du Scénario 2 à l'instant 4.1s : (a) VT, CDL et Msg du* V_5.

dans celle de V_2 entre 2.5 et 3s. A ces instants, la masse sur l'existence de la carte locale est vide, V_1 et V_2 n'ont pas détecté V_3. Tous les véhicules gardent V_3 dans leur carte distribuée quand les autres le détectent. Cette figure montre la différence entre la carte distribuée et la carte locale. La carte locale se limite à ce que détectent les capteurs tandis que la carte distribuée permet aux véhicules d'avoir une vue globale sur la scène.

Comparaison des CDD et CDL dans différentes situations

Pour chaque situation, nous comparons pour chaque véhicule les VP, FP et FN, les calculs sur la carte locale et distribuée (figure 4.23 a, b et c). Nous calculons aussi la précision et le rappel (tableau 4.1).

Nous avons testé le cas où les véhicules reçoivent tous les messages envoyés. Les figures 4.23a, 4.23b et 4.23c montre les détections de V_0, V_1 et V_2. L'échange des cartes distribuées augmente le nombre de bonnes détections (VP) des véhicules. Les mauvaises détections (FN) restantes sont dues au fait que les véhicules peuvent garder des fausses pistes dans leur carte distribuée pendant un certain temps. Le tableau 4.1 montre la précision et le rappel pour les cartes locales et distribuées des trois véhicules. Nous remarquons que le taux de rappel de la CD augmente considérablement par rapport à la CL. En revanche, à cause de quelques fausses pistes provoquées par l'association, la précision décroît légèrement.

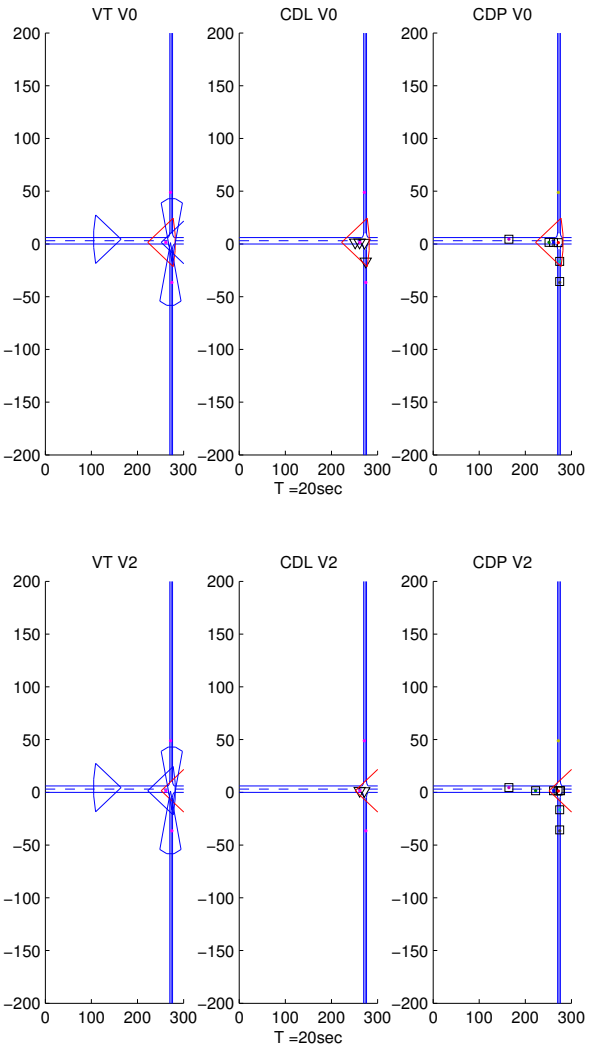


FIGURE ***4.19.:*** *Résultats du Scénario 2 à l'instant 20s : (a) VT, CDL et CDP du V_0, (b) VT, CDL et CDP du V_2.*

TABLE ***4.1.:*** *Précision et rappel dans le cas où les véhicules reçoivent tous les messages envoyés*

	V_0		V_1		V_2	
	Précision	Rappel	Précision	Rappel	Précision	Rappel
CL	1	0.59	1	0.34	1	0.34
CD	0.94	0.79	0.98	0.81	0.99	0.83

Cas d'un capteur défectueux : Dans cette simulation, V_0 ne détecte pas V_3 entre les instants 1.9 et 5.3s. Dans ce cas, nous cherchons à évaluer l'influence de la

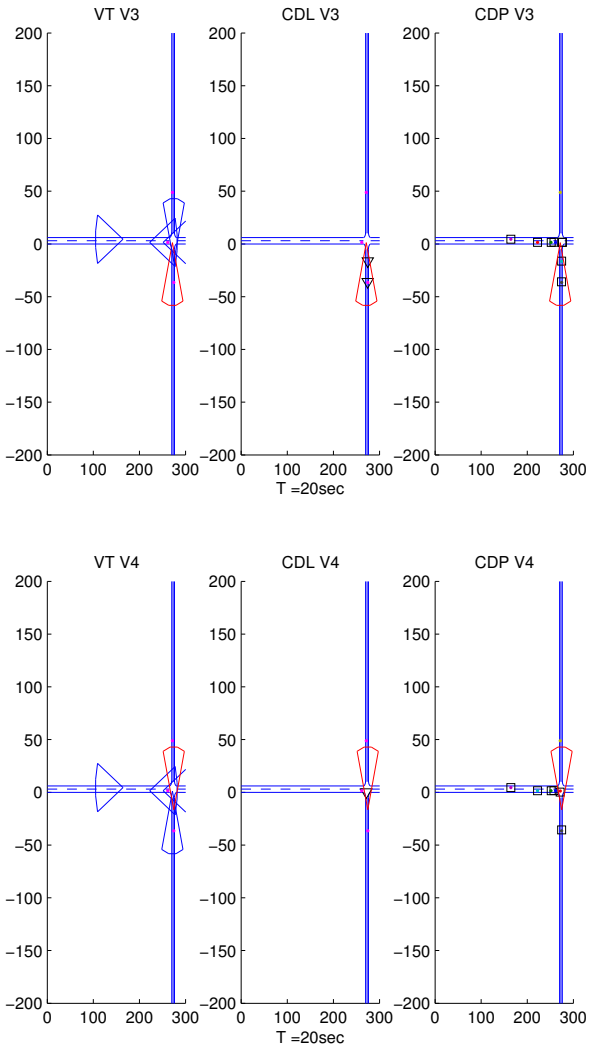


FIGURE *4.20.: Résultats du Scénario 2 à l'instant 20s : (a) VT, CDL et CDP du V_3, (b) VT, CDL et CDP du V_4.*

non-détection d'un capteur sur l'ensemble. Nous remarquons dans la figure 4.24a l'augmentation de la non détection de V_0 dans sa carte locale. L'effet est plus important sur le taux de FP de V_0. Le tableau 4.2 montre la légère variation au niveau du rappel de la CL et de la précision de la CD de V_0.

Cas d'une antenne wifi défectueuse : Cet exemple montre le cas où V_2 a un problème avec son antenne wifi. Il ne reçoit plus les messages des autres véhicules pendant 7.5s. La figure 4.25c montre l'influence de ce problème sur la diminution de la bonne détection de V_2 (VP) et l'augmentation du taux de non-détection (FP). Le tableau 4.3 montre l'influence de ce problème sur le rappel de V_2.

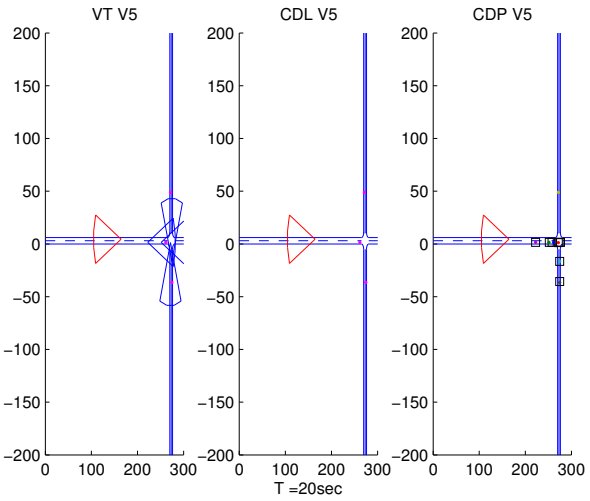


FIGURE 4.21.: *Résultats du Scénario 2 à l'instant 20s : (a) VT, CDL et Msg du V_5.*

TABLE 4.2.: *Précision et rappel dans le cas d'un capteur défectueux*

	V_0		V_1		V_2	
	Précision	Rappel	Précision	Rappel	Précision	Rappel
CL	1	0.53	1	0.34	1	0.34
CD	1	0.79	1	0.81	0.99	0.81

TABLE 4.3.: *Précision et rappel dans le cas d'une antenne wifi défectueuse*

	V_0		V_1		V_2	
	Précision	Rappel	Précision	Rappel	Précision	Rappel
CL	1	0.59	1	0.34	1	0.34
CD	0.93	0.73	0.97	0.77	0.97	0.57

Changement de portée de communication : Le changement de la portée de communication a une influence sur la connaissance distribuée des véhicules. Les portées de communication ont été changées de la manière suivante :

– portée 1 : c'est le cas des figures 4.26a, 4.26b et 4.26c. Les véhicules ont une faible portée, de telle sorte que les véhicules V_0 et V_1, qui sont très proches, ne peuvent pas communiquer. Ceci est visible en comparant les détections de V_0 pour la carte distribuée et locale : les valeurs sont les mêmes. Ce phénomène

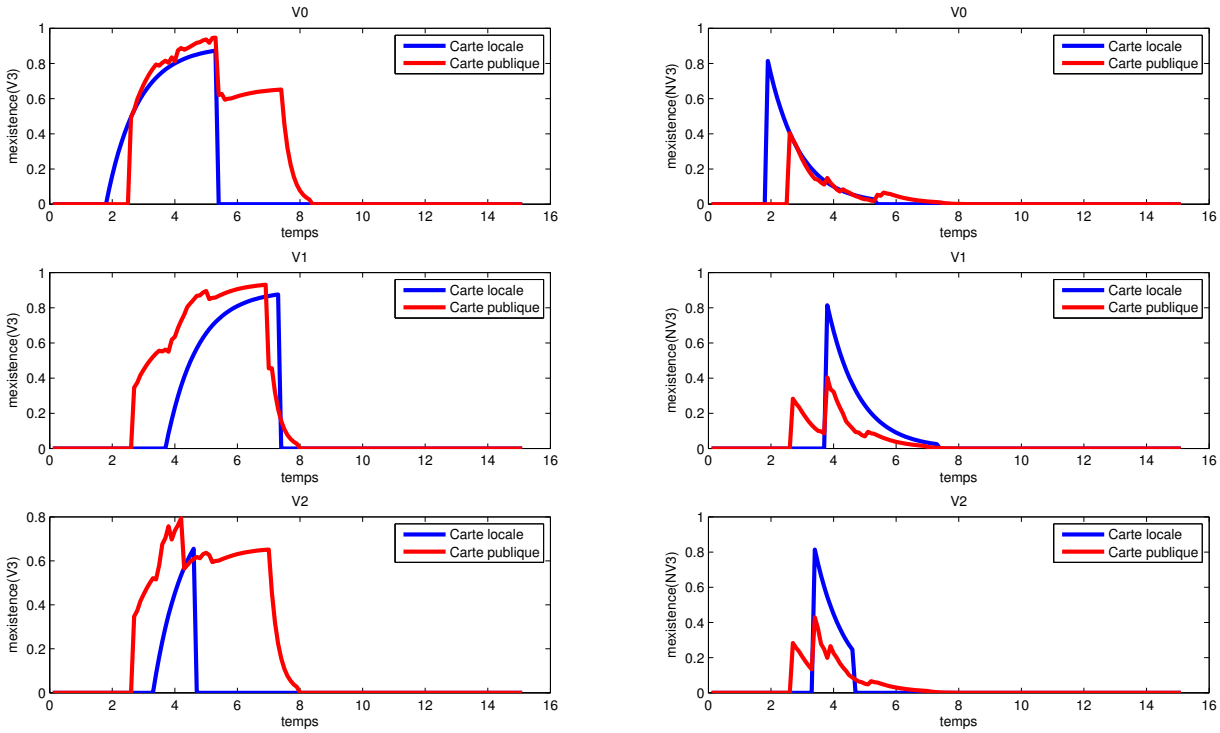


FIGURE 4.22.: *La masse sur l'existence du véhicule 3 détecté par les 3 autres véhicules. Les courbes bleues représentent la masse relative à l'existence dans la carte locale et les rouges celles de la carte distribuée de chaque véhicule. La première colonne représente $m(V_3)$ et la deuxième colonne montre $m(NV_3)$ où NV_3 correspond à $\overline{V_3}$*

est dû au fait que V_0 ne reçoit pas de messages. Les taux de non-détection de V_1 et V_2 augmentent aussi.

– portée 2 : la portée est augmentée de telle façon que les véhicules V_0 et V_1 puissent toujours communiquer et V_3 peut échanger sa carte quand il s'approche d'eux à la même portée (figures 4.26d, 4.26e et 4.26f).
– portée 3 : la portée 3 est plus grande que les deux premières portées, ce qui correspond au cas où tous les véhicules reçoivent tous les messages envoyés (figures 4.26g, 4.26h et 4.26i).

Le tableau 4.4 montre l'influence du changement de la portée sur la performance de la méthode dans ce scénario. Nous remarquons l'augmentation du rappel de chaque véhicule quand la portée augmente et une faible variation de la précision.

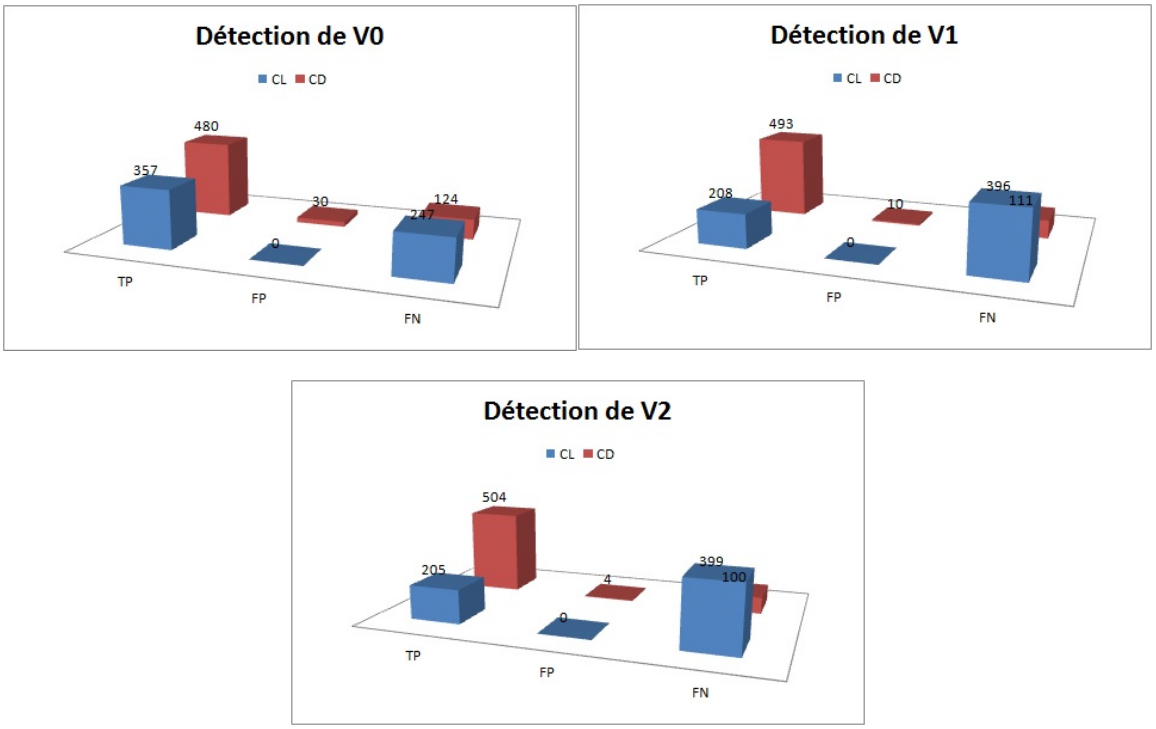


FIGURE *4.23.: Comparaison de la carte locale et distribuée dans le cas d'une communication parfaite.*

TABLE *4.4.: Précision et rappel dans le cas de changement de portée*

		V_0		V_1		V_2	
		Précision	Rappel	Précision	Rappel	Précision	Rappel
CL	portée 1	1	0.59	1	0.34	1	0.34
	portée 2	1	0.59	1	0.34	1	0.34
	portée 3	1	0.59	1	0.34	1	0.34
CD	portée 1	1	0.59	1	0.34	1	0.4
	portée 2	0.98	0.83	1	0.64	0.98	0.47
	portée 3	0.94	0.79	0.98	0.81	0.99	0.83

Envoyer sa CDL ou sa CDD ? : Comme mentionné dans la section 4.1, les véhicules peuvent envoyer soit leurs propres détections, soit les données reçues par les autres telles quelles ou combinées avec leurs propres données. Dans cette partie, nous comparons le changement de type des données envoyées dans les messages. Deux types de messages sont envoyés : les véhicules envoient dans un premier temps des messages contenant leurs cartes locales et dans un second temps des

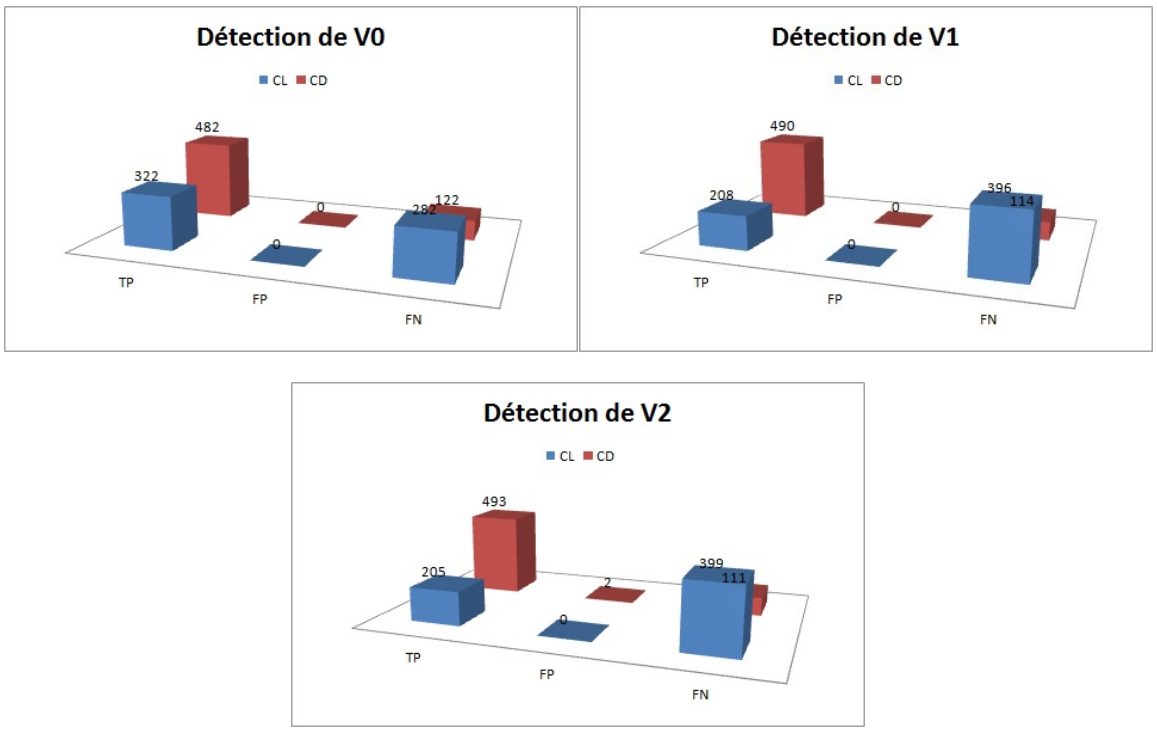


FIGURE 4.24.: *Comparaison de la carte locale et distribuée dans le cas d'un capteur défectueux.*

messages contenant leurs cartes distribuées. La figure 4.27 montre la différence entre ces deux cas. L'échange de la carte distribuée permet d'augmenter le taux de bonne détection VP pour les véhicules, et provoque bien l'augmentation du champ de perception des véhicules. Un message contenant la CL est limité par le champ de vision du capteur alors qu'un message contenant la CD permet au véhicule récepteur d'ajouter la plupart des objets détectés dans la scène. Ce phénomène est mis en évidence dans le tableau 4.5 qui montre l'augmentation du rappel dans le cas de l'envoi d'un message contenant la CD.

4.6. Conclusion

Dans ce chapitre, nous avons présenté une seconde application de l'algorithme de fusion distribuée. Cette application se base sur le principe de la perception coopérative, qui suppose que les véhicules coopèrent pour améliorer leur champ de perception. Chaque véhicule est équipé de capteurs lui permettant de détecter les objets dans son environnement proche et d'une antenne wifi pour communiquer

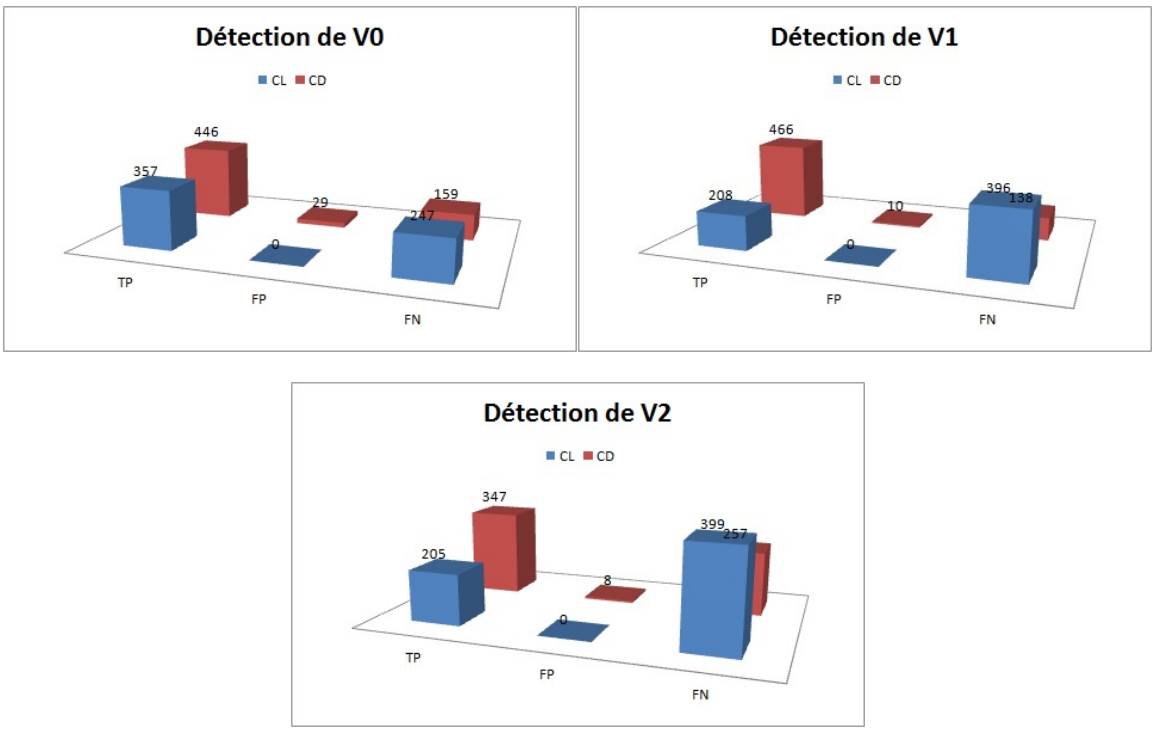


FIGURE 4.25.: *Comparaison de la carte locale et distribuée dans le cas d'une antenne wifi défectueuse.*

TABLE 4.5.: *Précision et rappel pour l'envoi des messages contenant la carte locale et distribuée*

		V_0		V_1		V_2	
		Précision	Rappel	Précision	Rappel	Précision	Rappel
CL		1	0.59	1	0.34	1	0.34
CD	Msg CL	0.98	0.63	0.99	0.58	0.98	0.45
	Msg CD	0.98	0.83	1	0.64	0.98	0.47

avec les autres véhicules. L'algorithme de fusion distribuée permet de construire une carte de l'environnement dynamique comprenant les objets dans le champ de vision du capteur ainsi que ceux envoyés par les autres véhicules. La fusion distribuée combine les confiances dans l'existence des objets à l'aide d'opérateurs appropriés selon la source des données. Il en résulte une carte dynamique distribuée offrant une perception augmentée de l'environnement. La mise en oeuvre d'une telle application nécessite de gérer les recalages temporel et spatial des données échangées ainsi la mise en correspondance des objets par un algorithme d'associa-

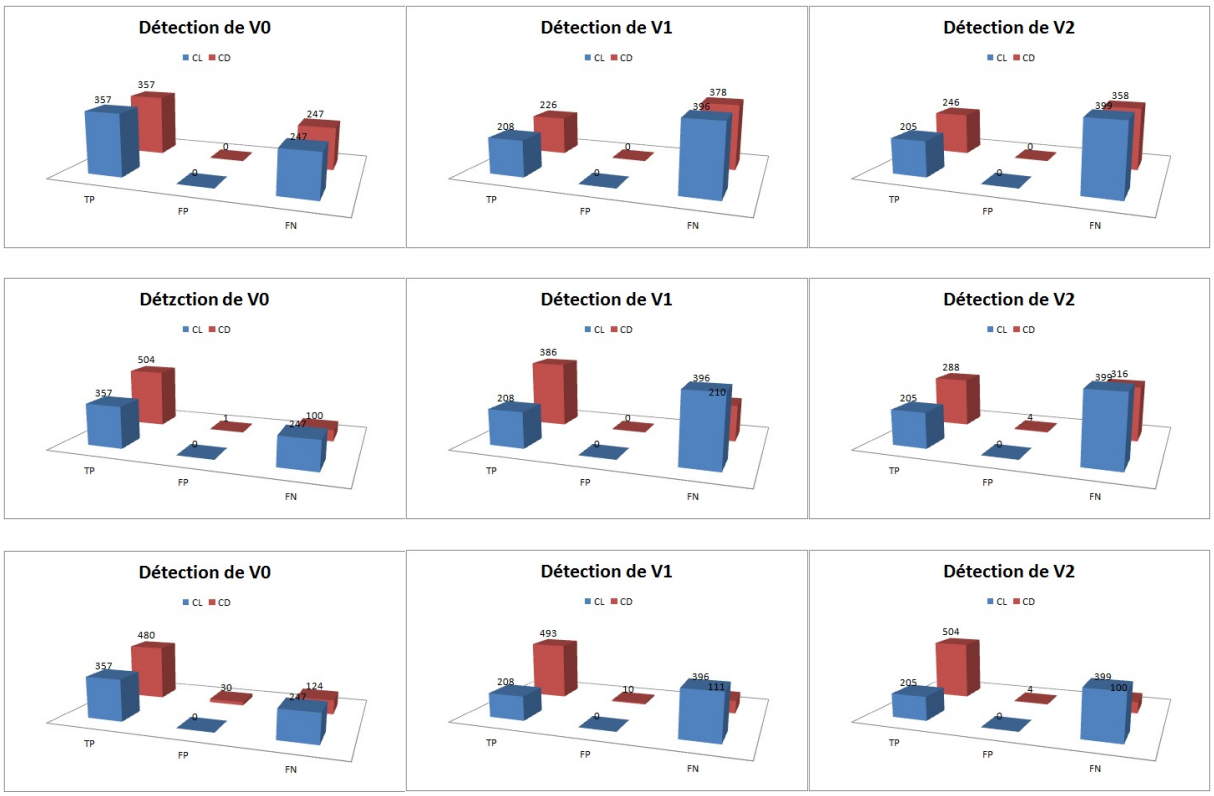


FIGURE 4.26.: *Comparaison de la carte locale et distribuée dans le cas de changement de portée : Portée 1 : figures a, b et c. Portée 2 : figures d, e et f. Portée 3 : figures f, g et h.*

tion qui sera décrit dans le chapitre suivant. La construction de la carte dynamique distribuée a été validée par simulation sur différents scénarios de situation routière impliquant plusieurs véhicules.

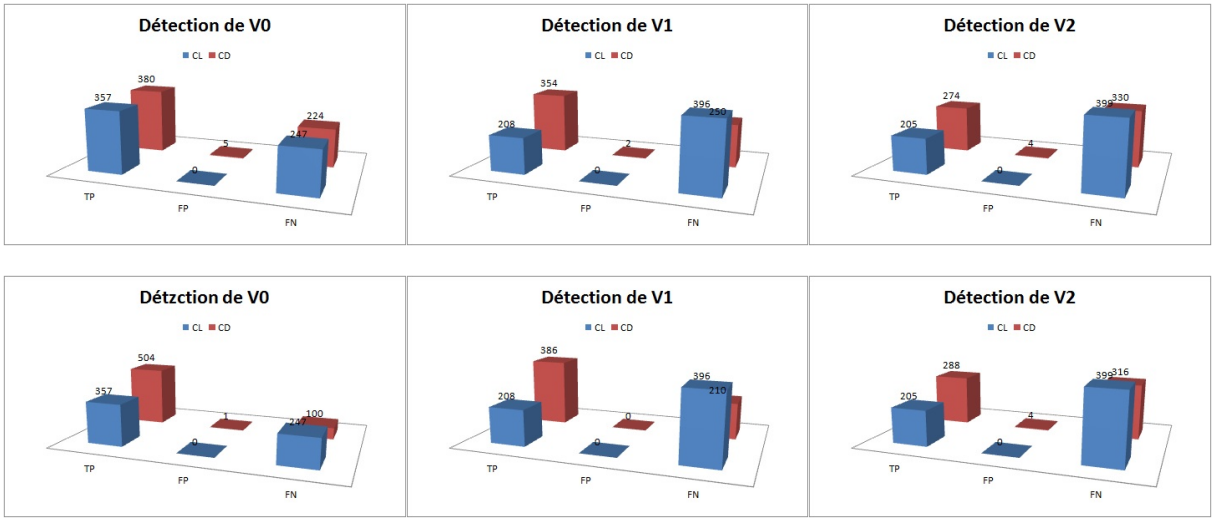


FIGURE 4.27.: *Comparaison des détections des véhicules pour une même portée, dans le cas où le message contient la connaissance locale : a) V_0, b) V_1 et c) V_2 et dans le cas où le message contient la connaissance publique : d) V_0, e) V_1 et f) V_2*

CHAPITRE 5

Association optimale d'objets

5.1. Introduction

L'association d'objets est une étape préalable dans un processus de fusion de données. Dans le cadre de notre approche de fusion distribuée, elle intervient dans l'application de la carte dynamique distribuée. A la réception des messages, le véhicule doit associer la carte qu'il reçoit avec sa propre carte distribuée, et le résultat avec sa carte locale. Les objets sont caractérisés par les informations cinématiques (position, vitesse) et les informations d'identification (classe). Dans l'application de détection d'attaque sybil, les émetteurs sont caractérisés par leur identifiant et l'étape d'association n'est donc pas nécessaire.

L'association d'objets est un problème difficile car le nombre d'objets dans une scène est inconnu et les données des capteurs peuvent être incomplètes et incertaines (fausse alarme et non détection). Nous écarterons les méthodes dites denses ou semi-denses comme, par exemple, la méthode de Ransac, qui ont été développées pour la mise en correspondance de points caractéristiques entre deux images. Ces méthodes sont performantes lorsque le nombre des primitives est élevé, ce qui n'est pas souvent le cas lorsqu'on veut suivre des véhicules et des piétons dans une scène routière.

Différents travaux de recherche ont étudié ce problème afin d'associer les objets provenant des observations multiples détectées par différents capteurs ou bien par un seul capteur à des instants différents. La première situation est rencontrée, par exemple, en suivant différents objets comme des cibles [BSF88] ou des cellules

d'orage [EDG90]. La seconde situation se produit dans le cadre des applications multi-capteurs, comme la surveillance audio-vidéo ou les surveillances des scènes routières pour l'aide à la conduite [HBCB12]. Les méthodes classiques d'associations ont été discutées en détail dans plusieurs ouvrages dont les plus connus sont les livres de Bar-Shalom et Fortmann [BSF88] et de Blackman et Popoli [BP99]. Les principales méthodes sont les algorithmes NN (Nearest Neighbor), GNN (Global Nearest Neighnor), PDA (Probabilistic Data Association), JPDA (Joint Probabilistic Data Association) et MHT (Multi Hypothesis Tracking). Elles ont été développées essentiellement pour des applications militaires de type radar. Ces différentes méthodes et d'autres ([Smi06],[SAR+12]), diffèrent par leur complexité, leur capacité à gérer les incertitudes et les ambiguïtés dans les associations. Toutes ces méthodes ont été testées et implémentées dans des systèmes réels.

Dans sa thèse, Bailey [Bai02] a décomposé le problème d'associations de données pour le pistage en deux sous-problèmes :

- Le premier problème consiste à réduire les ambiguïtés des associations c'est-à-dire à limiter les observations potentiellement candidates à l'association avec une piste. Ainsi, si à chaque piste une et une seule observation est candidate et si celle-ci n'est associée à aucune autre piste, alors l'association ne pose pas de problème. Les méthodes de fenêtrage ou d'associations par plus proche voisin (NN, GNN) apportent des solutions à ce problème.
- Le second problème consiste à résoudre les ambiguïtés qui persistent. Les méthodes PDA et JPDA consistent à développer des filtres de mise à jour des pistes en prenant en compte plusieurs observations. La méthode MHT développe plusieurs hypothèses d'association en parallèle et choisit la solution la plus probable dans la phase de décision.

Dans ce chapitre, nous proposons une formalisation du problème d'association dans le cadre des théories de fonctions de croyance pour modéliser les incertitudes liées au capteur et les problèmes d'occultation. Notre méthode est basée sur la recherche de la relation la plus plausible entre deux ensembles d'objets. Des travaux reposant sur le cadre des fonctions de croyances ont été développés. Dans [AS01], les auteurs présentent une méthode qui détermine la présence des objets observés par différents capteurs. Ils utilisent le degré de conflit pour résoudre le problème d'associations de données et appliquent leur procédure à la détection de sous-marins. Une méthode d'association de données pour le suivi de cibles multiples basée sur les fonctions de croyance a été proposée dans [MACC05]. Schubert dans

[Sch00] gère l'incohérence des informations comme une étape préalable à la fusion d'informations. Différentes techniques ont été proposées pour l'association des objets perçus avec les objets connus pour les applications de suivi d'objets. Gruyer et Cherfaoui [GC99], à la suite de Rombaut et Cherfaoui [RC97], ont développé un algorithme d'association multi-objets. Pour prendre une décision, ils fournissent une bonne solution pour le suivi multi-objets. Mercier et al. [MLJ11], prolongeant les travaux de Rombaut [Rom98] et Gruyer et al. [GRLD03], ont développé une méthode qui traite l'apparition et la disparition des objets. Ils associent les objets détectés à l'instant t avec les objets connus (pistes) à l'instant précédent. Ristic et Smets [RS07] proposent une méthode pour associer les objets en utilisant l'information sur les classes des objets.

Dans ce chapitre, nous présentons une méthode d'association de deux ensembles d'objets en utilisant tous les attributs fournis par les capteurs. Cette méthode se base sur la recherche de la relation la plus plausible entre les objets. La méthode est validée expérimentalement sur des données simulées et réelles.

5.2. Relation entre deux ensembles d'objets

Le problème d'association consiste à trouver une relation entre deux ensembles finis d'objets $E = \{e_1, \ldots, e_n\}$ et $F = \{f_1, \ldots, f_p\}$ de cardinalités non nécessairement égales [BSF88], [BP99]. Nous faisons l'hypothèse que chaque objet d'un ensemble est associé au plus à un seul objet de l'autre ensemble. Un objet de E peut ne pas être associé à un objet de F, du fait de la disparition de l'objet entre deux instants successifs ou de la non détection par un des capteurs. Mathématiquement, nous cherchons une relation $R \subseteq E \times F$ telle que, pour tout i, j et k :

$$(e_i, f_j) \in R \text{ et } (e_i, f_k) \in R \Rightarrow j = k \tag{5.1a}$$

et

$$(e_i, f_k) \in R \text{ et } (e_j, f_k) \in R \Rightarrow i = j. \tag{5.1b}$$

Une telle relation peut être représentée par une matrice de dimension (n, p) telle $R_{ij} = 1$ si $(e_i, f_j) \in R$ et $R_{ij} = 0$ sinon. Le problème d'association est formalisé dans le cadre des fonctions de croyance. Nous considérons qu'on reçoit des éléments d'évidence sur les associations possibles de chaque paire (e_i, f_j). Mathématique-

ment, chaque élément d'évidence peut être représentée par une fonction de masse $m_{i,j}$ sur le cadre de discernement $\Theta_{i,j} = \{0, 1\}$, de telle sorte que $m_{ij}(\{1\}) = \alpha_{i,j}$ est la probabilité que $R_{ij} = 1$, $m_{ij}(\{0\}) = \beta_{i,j}$ est la probabilité que $R_{ij} = 0$ et $m_{ij}(\{0, 1\}) = 1 - \alpha_{i,j} - \beta_{i,j}$ est la probabilité de ne rien savoir sur R_{ij}. Il s'agit de trouver la meilleure relation R^* dans l'ensemble $\mathcal{R}$ des relations vérifiant (5.1a)-(5.1b). La recherche d'une relation dans $\mathcal{R}$ en se basant sur une paire de fonctions de masses a été abordée par différents auteurs (exemple [RC97]-[DOO13]). Cependant, seules les solutions heuristiques ont été obtenues jusqu'à présent. La solution la plus élaborée, proposée par Mercier et al. [MLJ11], consiste à combiner en premier lieu les fonctions de masse $m_{ij}{}^p$ pour chaque i, et à trouver la relation R en maximisant la probabilité pignistique [SK94]. Cependant, cet algorithme nécessite l'énumération de tous les éléments de R, ce qui devient irréalisable quand n et p dépassent quelques unités. En plus, cette méthode manque une propriété de symétrie fondamentale, car elle peut donner deux résultats différents si E et F sont intervertis.

5.3. Méthode de Mercier et al.

La méthode développée par Mercier et al. [MLJ11] permet d'associer les objets perçus avec les objets connus à l'instant précédent. Ils modélisent le problème d'association dans le cadre des fonctions de croyance. La méthode est basée sur celles de Rombaut [Rom98] et de Gruyer et al. [GRLD03]. Deux points de vues différents sont considérés : le point de vue des objets perçus et celui des objets connus. Nous allons décrire dans ce qui suit leurs propres notations :

- e_i représente un objet perçu à l'instant t, $i \in I$={1, . . . , N}, N est le nombre d'objets perçus à cet instant ;
- f_i représente un objet connu à l'instant précédent $t - 1$, $j \in J$={1, . . . , M}, M est le nombre d'objets connus à $t - 1$;
- $\star$ représente l'absence d'un objet.

Le but est de trouver la meilleure association possible entre les objets perçus $\{e_1, e_2, \ldots, e_N, \star\}$ et les objets connus $\{f_1, f_2, \ldots, f_M, \star\}$ sous les contraintes suivantes :

- chaque objet perçu e_i doit être associé avec au plus un seul objet connu ;
- chaque objet connu f_j doit être associé avec au plus un objet perçu ;

– la proposition $\star$ peut être associée avec n'importe quel objet.

Le cadre de discernement est défini de la façon suivante :

– $\Omega_{i,j} = \{\mathcal{R}_{i,j}, \overline{\mathcal{R}_{i,j}}\}$: représente le fait que l'objet e_i est associé à f_j ou non ;
– $\Omega_{e_i} = \{f_1, f_2, \ldots, f_M, ^\star\}$: est l'ensemble des objets susceptibles d'être associés à e_i. Le symbole $\star$ signifie que e_i est nouvellement apparu ;
– $\Omega_{f_j} = \{e_1, e_2, \ldots, e_N, ^\star\}$: est l'ensemble d'éventuels objets associés à f_j. Le symbole $\star$ signifie que f_j a disparu ou est caché.

L'information est représentée par des fonctions de masse $m^{\Omega_{i,j}}$ dans le cadre $\Omega_{i,j}$, $i \in I$, $j \in J$:

– $m^{\Omega_{i,j}}(\mathcal{R}_{ij})$ représente l'association entre les objets e_i et f_j ;
– $m^{\Omega_{i,j}}(\overline{\mathcal{R}_{ij}})$ exprime le fait que e_i n'est pas associé à f_j ;
– $m^{\Omega_{i,j}}(\Omega_{i,j})$ représente l'ignorance.

L'information est représentée dans un cadre commun global (extension vide) :

$$m^{\Omega_{i,j}\uparrow\Omega_{e_i}(\rho_{i,j}(A))} = m^{\Omega_{i,j}}(A), \forall A \subseteq \Omega_{i,j}, \tag{5.2}$$

où $\rho_{i,j}$ est le raffinement de Ω_{ij} sur Ω_{e_i} et défini par $\rho_{i,j}(\mathfrak{R}_{\mathrm{i,j}}) = \{f_j\}$ et $\rho_{i,j}(\overline{\mathfrak{R}_{\mathrm{i,j}}}) = \overline{\{f_j\}}$. Par conséquent, pour tout $(i,j) \in I \times J$:

$$\begin{array}{rcl} m^{\Omega_{e_i}}(\{f_j\}) & = & m^{\Omega_{i,j}}(\mathcal{R}_{ij}), \\ m^{\Omega_{e_i}}(\overline{\{f_j\}}) & = & m^{\Omega_{i,j}}(\overline{\mathcal{R}_{ij}}), \\ m^{\Omega_{e_i}}(\Omega_{e_i}) & = & m^{\Omega_{i,j}}(\Omega_{i,j}). \end{array} \tag{5.3}$$

De la même façon, la fonction de masse $m^{\Omega_{i,j}}$ peut être exprimée dans Ω_{f_j} :

$$\begin{array}{rcl} m^{\Omega_{f_j}}(\{e_i\}) & = & m^{\Omega_{i,j}}(\mathcal{R}_{ij}), \\ m^{\Omega_{f_j}}(\overline{\{e_i\}}) & = & m^{\Omega_{i,j}}(\overline{\mathcal{R}_{ij}}), \\ m^{\Omega_{f_j}}(\Omega_{f_j}) & = & m^{\Omega_{i,j}}(\Omega_{i,j}). \end{array} \tag{5.4}$$

L' approche de Mercier comprend les étapes suivantes :

– pour chaque e_i, on calcule M fonctions de masse $m^{\Omega_{e_i}}$ concernant l'association de chaque objet X_i avec Y_j. Ces masses sont combinées avec la règle de combinaison conjonctive et la décision est prise en calculant la probabilité pignistique $BetP^{\Omega_{e_i}}$.
– pour chaque f_j, N fonctions de masses $m^{\Omega_{f_j}}$ sont calculées pour exprimer l'association de chaque objet Y_j avec X_i. Ces masses sont combinées avec la règle de combinaison conjonctive et de la même façon, la décision est prise en cal-

culant la probabilité pignistique $BetP^{\Omega_{f_j}}$.

Finalement, on obtient deux tableaux présentant la probabilité pignisitique des deux points de vue : objets connus et objets perçus. Les auteurs remarquent que le coût de la combinaison et du raffinement dans un cadre commun est très élevé. Donc, ils proposent de raisonner sur les tableaux de probabilités pignistiques. L'algorithme de décision proposé par les auteurs, considère du point de vue des objets perçus $(M+1)^N$ associations possibles. Ils suppriment de cet ensemble d'associations celles qui ne satisfont pas les contraintes et calculent le produit des probabilités pignistiques sur ces ensembles. Ils choisissent le maximum comme une solution de l'association du point de vue des objets perçus. Ce même algorithme est appliqué du point de vue des objets connus. L'un des problèmes de cette méthode est que les deux points de vue peuvent conduire à des décisions différentes. La solution proposée consiste à prendre une décision en privilégiant soit les objets perçus, soit les objets connus. Cette méthode ne vérifie donc pas une propriété de symétrie fondamentale et l'énumération des $(M+1)^N$ associations possibles rend la méthode coûteuse et parfois inutilisable.

5.4. Formalisme d'association d'objets

Dans cette section, nous montrons la recherche de la relation la plus plausible entre deux ensembles d'objets, sur la base d'éléments d'évidence indépendants relatifs à chaque paire d'objets, peut être formalisé comme un problème de programmation linéaire en variables binaires.

5.4.1. Formalisation du problème

Comme expliqué dans la section 5.2, nous considérons que les informations disponibles sur l'association entre deux ensembles E et F sont représentées par np fonctions de masse $m_{i,j}$, $1 \leq i \leq n$, $1 \leq j \leq p$. Chaque $m_{i,j}$ représente un élément d'évidence sur la variable binaire $R_{i,j}$ qui est égale à 1 si e_i et f_j correspondent à la même entité et 0 sinon. Nous notons que l'absence d'informations sur l'association entre e_i et f_j peut être représentée par une fonction de masse vide $m_{i,j}(\{0,1\}) = 1$. Plus précisément, $m_{i,j}$ est basée sur la mesure de similarité entre différents attributs caractérisant l'objet. Les exemples seront détaillés dans les sections suivantes.

L'idée principale de notre approche est d'exprimer toutes les évidences disponibles dans le cadre de discernement $\mathcal{R}$ défini comme l'ensemble de toutes les associations possibles entre E et F vérifiant les contraintes (5.1a)-(5.1b). Compte tenu de l'indépendance des np éléments de l'évidence, les np fonctions de masse $m_{i,j}$ peuvent être combinées par la règle de Dempster et la plausibilité de n'importe quelle relation $R \in \mathcal{R}$ peut être simplement calculée en utilisant la fonction de contour de plausibilité :

$$(pl_1 \oplus pl_2)(\theta) = \frac{pl_1(\theta) pl_2(\theta)}{1-\kappa}, \tag{5.5}$$

pour tout $\theta \in \Theta$.

Soit $\mathcal{R}_{ij}$ l'ensemble des relations possibles où e_i et f_j sont associés :

$$\mathcal{R}_{ij} = \{R \in \mathcal{R} | R_{ij} = 1\}. \tag{5.6}$$

Chaque fonction de masse m_{ij} sur $\Theta_{ij} = \{0,1\}$ peut être exprimée dans $\mathcal{R}$ en affectant la masse $m_{ij}(\{1\}) = \alpha_{ij}$ à $\mathcal{R}_{ij}$, $m_{ij}(\{0\}) = \beta_{ij}$ à $\overline{\mathcal{R}_{ij}}$et $m_{ij}(\{0,1\}) = 1 - \alpha_{ij} - \beta_{ij}$ à $\mathcal{R}$, où $\overline{\mathcal{R}_{ij}}$ est le complément de $\mathcal{R}_{ij}$. Soit pl_{ij} la fonction de contour correspondante. Elle peut être exprimée sous la forme suivante :

$$pl_{ij}(R) = \begin{cases} 1-\beta_{ij} & \text{si } R \in \mathcal{R}_{ij}, \\ 1-\alpha_{ij} & \text{sinon}, \end{cases} \tag{5.7}$$

pour tout $R \in \mathcal{R}$, ce que l'on peut écrire de manière plus concise sous la forme suivante :

$$pl_{ij}(R) = (1-\beta_{ij})^{R_{ij}} (1-\alpha_{ij})^{1-R_{ij}}. \tag{5.8}$$

Soit m la fonction de masse sur $\mathcal{R}$ résultant de la combinaison des np masses en utilisant la règle de Dempster. A partir de (5.5), sa fonction de contour pl est proportionnelle au produit des np fonctions de masses pl_{ij} :

$$pl(R) \propto \prod_{i,j} (1-\beta_{ij})^{R_{ij}} (1-\alpha_{ij})^{1-R_{ij}}, \tag{5.9}$$

et le logarithme de la fonction de contour est :

$$\ln pl(R) = \sum_{i,j} \left[R_{ij} \ln(1-\beta_{ij}) + (1-R_{ij}) \ln(1-\alpha_{ij}) \right] + C, \tag{5.10}$$

où C est une constante et l'on suppose que $\beta_{ij} < 1$ et $\alpha_{ij} < 1$ pour tout i et j. Les cas où $\beta_{ij} = 1$ ou $\alpha_{ij} = 1$ (qui correspondent au cas où on est certain que e_i et f_j, doivent être associés ou non) peuvent être traités en prenant $\beta_{ij} = 1 - \epsilon$ ou $\alpha_{ij} = 1 - \epsilon$ pour une valeur arbitraire $\epsilon > 0$.

La relation la plus plausible R^* peut donc être trouvée en résolvant le problème d'optimisation linéaire en variables binaires suivant :

$$\max_R \sum_{i,j} w_{ij} R_{ij} = \max \ln pl(R)$$

sous les contraintes (5.1a)-(5.1b), avec

$$w_{ij} = \ln \frac{1 - \beta_{ij}}{1 - \alpha_{ij}}. \tag{5.11}$$

Nous pouvons observer qu'une connaissance a priori sur l'association des objets peut être facilement intégrée dans ce cadre. Par exemple, supposons que les relations qui associent plus d'objets sont considérés *a priori* comme plus plausibles ou, au contraire, moins plausibles. Une telle connaissance peut être représentée par une fonction de croyance avec la fonction contour pl_0 tel que :

$$pl_0(R) \propto \exp\left(\lambda \sum_{i,j} R_{ij}\right), \tag{5.12}$$

où λ est un paramètre scalaire. Une valeur positive (respectivement, négative) de λ favorise les relations R de cardinal supérieur (respectivement, inférieur). En combinant cette fonction de contour avec les fonctions de contour pl_{ij} par la règle de Dempster et en prenant le logarithme, nous obtenons

$$\ln pl(R) \propto \sum_{i,j} [R_{ij} \ln(1 - \beta_{ij}) + (1 - R_{ij}) \ln(1 - \alpha_{ij}) + \lambda R_{ij}] + C. \tag{5.13}$$

Le problème de maximisation obtenu a alors la même forme que ci-dessus avec w_{ij} défini maintenant par

$$w_{ij} = \lambda + \ln \frac{1 - \beta_{ij}}{1 - \alpha_{ij}}. \tag{5.14}$$

5.4.2. La résolution du problème et l'analyse de complexité

Le problème d'association optimale formalisé dans le chapitre 5 et noté P dans ce qui suit, peut être formulé sous la forme du programme linéaire en nombres entiers suivant :

$$\max \sum_{i=1}^{n} \sum_{j=1}^{p} w_{ij} R_{ij} \tag{5.15}$$

sous

$$\sum_{j=1}^{p} R_{ij} \leq 1 \qquad \forall i \in \{1, \ldots, n\} \tag{5.16a}$$

$$\sum_{i=1}^{n} R_{ij} \leq 1 \qquad \forall j \in \{1, \ldots, p\} \tag{5.16b}$$

$$R_{ij} \in \{0, 1\} \quad \forall i \in \{1, \ldots, n\}, \forall j \in \{1, \ldots, p\}, \tag{5.16c}$$

où les contraintes (5.16a) et (5.16b) sont liées respectivement aux équations (5.1a) et (5.1b).

Nous considérons que $n \geq p$ (Si ce n'est pas le cas, E et F sont échangés). Afin de trouver une solution, P est réduit au problème P' suivant :

$$\max \sum_{i=1}^{n} \sum_{j=1}^{n} w'_{ij} R'_{ij} \tag{5.17}$$

sous

$$\sum_{j=1}^{n} R'_{ij} = 1 \qquad \forall i \in \{1, \ldots, n\} \tag{5.18a}$$

$$\sum_{i=1}^{n} R'_{ij} = 1 \qquad \forall j \in \{1, \ldots, n\} \tag{5.18b}$$

$$R'_{ij} \in \{0, 1\} \quad \forall (i, j) \in \{1, \ldots, n\}^2, \tag{5.18c}$$

où $\forall i \in \{1, \ldots, n\}$, $\forall j \in \{1, \ldots, p\}$, $w'_{ij} = \max(0, w_{ij})$ et $\forall i \in \{1, \ldots, n\}$, $\forall j \in \{p+1, \ldots, n\}$, $w'_{ij} = 0$.

Proposition 1. *Soient R une solution optimale du problème P et R' une solution optimale du problème P', alors $\sum_{i=1}^{n} \sum_{j=1}^{n} w'_{ij} R'_{ij} = \sum_{i=1}^{n} \sum_{j=1}^{p} w_{ij} R_{ij}$.*

Preuve: *Supposons que* $\sum_{i=1}^{n}\sum_{j=1}^{n} w'_{ij}R'_{ij} < \sum_{i=1}^{n}\sum_{j=1}^{p} w_{ij}R_{ij}$. *Une nouvelle solution* R'' *pour le problème* P' *est extraite de la solution* R *de la manière suivante. Tout d'abord,* R'' *est initialisée avec* $R''_{ij} = 0, \forall (i,j) \in \{1,\ldots,n\}^2$.

Pour chaque $i \in \{1,\ldots,n\}$ *tel que* $\sum_{j=1}^{p} R_{ij} = 1$, R''_{ij} *est réglée à* R_{ij} *pour tout* $j \in \{1,\ldots,p\}$. *A ce stade d'élaboration de la solution* R'', *notons que* $\sum_{i=1}^{n}\sum_{j=1}^{n} w'_{ij}R''_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{p} w_{ij}R_{ij}$. *On note aussi que les contraintes (5.18a) et (5.18b) ne sont pas forcément conservées étant donné qu'un objet de* E *n'est pas nécessairement associé à un objet de* F *dans* R''. *Dans ce cas, il y a exactement le même nombre d'indices* $i \in \{1,\ldots,n\}$ *tel que* $\sum_{j=1}^{n} R''_{ij} = 0$ *que le nombre d'indices* $j \in \{1,\ldots,n\}$ *tel que* $\sum_{i=1}^{n} R''_{ij} = 0$. *Alors, chaque* $i \in \{1,\ldots,n\}$ *tel que* $\sum_{j=1}^{n} R''_{ij} = 0$ *est considéré de manière itérative. Nous cherchons le plus petit indice* j *avec* $\sum_{k=1}^{n} R''_{kj} = 0$ *(sachant que cet indice existe forcément) et* R''_{ij} *est mis à* 1. *Si* $j > p$, *alors* $w'_{ij} = 0$ *par définition du problème* P'. *Si* $j \leq p$, *alors forcément* $w'_{ij} = 0$. *En effet,* $w'_{ij} > 0 \Rightarrow w_{ij} > 0$ *et en mettant* R_{ij} *à* 1 *(il faut noter que cette nouvelle association peut être ajoutée à* R *étant donné que* $\sum_{k=1}^{p} R_{ik} = \sum_{k=1}^{n} R_{kj} = 0$*), nous obtenons une nouvelle solution du problème* P *avec un coût élevé, ce qui contredit que* R *est optimale. Ainsi, à la fin de chaque itération* $\sum_{i=1}^{n}\sum_{j=1}^{p} w'_{ij}R''_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{p} w_{ij}R_{ij}$ *tient toujours. Finalement,* R'' *est aussi une solution de* P' *avec* $\sum_{i=1}^{n}\sum_{j=1}^{p} w'_{ij}R''_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{p} w_{ij}R_{ij} > \sum_{i=1}^{n}\sum_{j=1}^{p} w'_{ij}R'_{ij}$ *ce qui contredit le fait que* R' *est une solution optimale de* P'.

Maintenant, supposons que $\sum_{i=1}^{n}\sum_{j=1}^{n} w'_{ij}R'_{ij} > \sum_{i=1}^{n}\sum_{j=1}^{p} w_{ij}R_{ij}$. *Une nouvelle solution* R'' *du problème* P *est extraite de la solution* R' *de la façon suivante. Pour tout* $i \in \{1,\ldots,n\}$ *et pour tout* $j \in \{1,\ldots,p\}$ R''_{ij} *est mise à* R'_{ij} *si* $w_{ij} \geq 0$ *et à* 0 *sinon. Notons que* R'' *est une solution de* P *puisque si les contraintes (5.18a) et (5.18b) sont respectées, alors les contraintes (5.16a) et (5.16b) sont aussi respectées. Puisque, par définition,* $w_{ij} < 0 \Rightarrow w'_{ij} = 0$ *alors* $\sum_{i=1}^{n}\sum_{j=1}^{p} w_{ij}R''_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{n} w'_{ij}R'_{ij} > \sum_{i=1}^{n}\sum_{j=1}^{p} w_{ij}R_{ij}$, *ce qui contredit le fait que* R *est une solution optimale de* P. □

Il devient évident qu'une solution optimale pour le problème P peut être obtenue en résolvant le problème P'. En effet, de n'importe quelle solution optimale R' du problème P', on peut construire une solution optimale R pour le problème P avec la même valeur de coût (et aussi optimale) avec $R_{ij} = R'_{ij}$ si $w_{ij} > 0$ et $R_{ij} = 0$ sinon. Il faut noter que la reconstitution du problème P' est de complexité $O(n^2)$ et la solution optimale R du problème P à partir d'une solution optimale de P' est de complexité $O(np)$. Donc, il reste à savoir comment résoudre le problème P' d'une manière efficace.

Dans la littérature de recherche opérationnelle, le problème P' est un *problème d'affectation* et peut être résolu avec une complexité $O(n^3)$ en utilisant *la méthode hongroise* [Kuh55] ou avec une complexité $O(n^{5/2}\log(n\max_{i,j} w_{ij}))$ en utilisant l'algorithme de Orlin et Ahuja [OA92]. Il faut noter, que dans la plupart du temps, les algorithmes utilisés pour résoudre le problème d'affectation sont décrits comme étant l'affectation de coût minimal avec des coûts entiers. Cependant, le problème de maximisation de coût P' peut être transformé en un problème de minimisation de coût en prenant $\max_{k,\ell} w_{k\ell} - w_{ij}$ à la place de w_{ij} et les coûts sont transformés en entiers en les multipliant par un nombre convenablement grand. En pratique, puisque le problème obtenu est un cas spécial du problème du chemin à coût minimal [AMO93], il est possible d'utiliser n'importe quel algorithme pour le résoudre. En particulier, il appartient à la classe des problèmes linéaires en nombre entier pour lesquels les matrices de contraintes sont unimodulaires et peut ainsi être résolu par des outils de programmation linéaire (sous les contraintes (5.6) de $R'_{ij} \in \{0,1\}$ à $R'_{ij} \in [0,1]$) sachant qu'il peut avoir toujours des solutions en nombre entier.

Example 1. *Afin d'illustrer comment un problème d'association d'objets peut être transformé en un problème d'affectation linéaire, nous allons considérer l'exemple suivant. Nous supposons que $n = 3$, $p = 4$, les α_{ij}s et β_{ij}s prennent les valeurs suivantes :*

$$(\alpha_{ij}) = \begin{pmatrix} 0.21 & 0.19 & 0.12 & 0.02 \\ 0.07 & 0.18 & 0.35 & 0.53 \\ 0.52 & 0.27 & 0.49 & 0.40 \end{pmatrix}, (\beta_{ij}) = \begin{pmatrix} 0.45 & 0.28 & 0.74 & 0.47 \\ 0.34 & 0.42 & 0.31 & 0.39 \\ 0.42 & 0.30 & 0.30 & 0.21 \end{pmatrix}.$$

La matrice de poids correspondante $W = (w_{ij})$ est :

$$W = \begin{pmatrix} -0.3621 & -0.1178 & -1.2192 & -0.6147 \\ -0.3429 & -0.3463 & 0.0597 & 0.2607 \\ 0.1892 & -0.0420 & 0.3167 & 0.2751 \end{pmatrix}.$$

En mettant les poids négatifs à 0 et en transformant W en une matrice carrée, nous obtenons

$$W' = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0597 & 0.2607 \\ 0.1892 & 0 & 0.3167 & 0.2751 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

En transformant W' à $\max_{k,\ell} w'_{k\ell} - W'$ afin de définir un problème de minimisation, nous avons :

$$W'' = \begin{pmatrix} 0.3167 & 0.3167 & 0.3167 & 0.3167 \\ 0.3167 & 0.3167 & 0.2570 & 0.0559 \\ 0.1274 & 0.3167 & 0 & 0.0416 \\ 0.3167 & 0.3167 & 0.3167 & 0.3167 \end{pmatrix}.$$

Le problème d'affectation linéaire avec la matrice de coût W'' a la solution suivante :

$$R' = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

En effaçant la dernière ligne et en prenant $R_{ij} = R'_{ij}$ si $w_{ij} \geq 0$ et $R_{ij} = 0$ sinon, nous obtenons la solution finale suivante :

$$R = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix},$$

ce qui veut dire que l'objet e_1 n'est pas associé, alors que e_2 est associé à f_4 et e_3 est associé à f_3.

5.5. Application à l'association d'objets

Dans cette partie, nous allons tout d'abord montrer comment les fonctions de masses peuvent être calculées à partir des attributs d'objets dans les applications de détection d'objets (5.5.1). Ensuite, nous présenterons les résultats expérimentaux sur des données simulées et réelles dans les applications de perception pour les véhicules intelligents.

5.5.1. Calcul des fonctions de masse

Les objets sont caractérisés par un ensemble d'attributs. Les valeurs de ces attributs pour chaque pair d'objets (e_i, f_j) sont considérés comme des éléments d'évidence pour construire la variable de l'association R_{ij}. La représentation de cette

information sous forme de masse m_{ij} dépend de la nature de l'attribut. Dans ce qui suit, nous allons considérer trois attributs parmi les plus utilisés dans les applications de détection d'objets : position, vitesse et classe.

Position

Soient E et F deux ensembles d'objets détectés par deux capteurs. On suppose que chaque capteur donne une position estimée pour chaque objet. Soit d_{ij} la distance entre les deux positions estimées de e_i et f_j ; il peut s'agir de la distance euclidienne ou de la distance de Mahalanobis si chaque capteur donne la matrice de covariance. La question qui se pose est : comment construire la fonction de masse m_{ij} à partir de la distance d_{ij} ? Il est évident qu'un seul objet ne peut avoir deux positions différentes et, inversement, deux objets ne peuvent pas occuper exactement la même position. En conséquence, une petite valeur de d_{ij} accrédite l'hypothèse $R_{ij} = 1$, tandis qu'une grande valeur de d_{ij} accrédite l'hypothèse $R_{ij} = 0$. Rappelons que $R_{ij} = 1$ signifie que les objets sont associés et $R_{ij} = 0$ représente le fait que les objets sont non associés. Une partie de la masse unitaire doit être affectée à $\Theta_{ij} = \{0, 1\}$ en prenant en compte la fiabilité du capteur. La fonction de masse m^p_{ij} sur la position prend alors la forme suivante :

$$m^p_{ij}(\{1\}) = \alpha\varphi(d_{ij}) \tag{5.19a}$$
$$m^p_{ij}(\{0\}) = \alpha\left(1 - \varphi(d_{ij})\right) \tag{5.19b}$$
$$m^p_{ij}(\Theta_{ij}) = 1 - \alpha, \tag{5.19c}$$

où $\alpha \in [0, 1]$ est le degré de confiance dans l'information du capteur et φ est une fonction décroissante à valeurs dans l'intervalle $[0, 1]$. Dans ce cas, φ est définie de la manière suivante :

$$\varphi(d) = \exp(-\gamma d), \tag{5.20}$$

où γ est un coefficient positif.

Vitesse

Supposons que chaque capteur retourne un vecteur vitesse pour chaque objet. Soit d'_{ij} la différence entre les vitesses des objets e_i et f_j. d'_{ij} est un élément d'évidence relatif à R_{ij}. Cependant, cette information n'a pas la même interprétation que

dans le cas précédent : ici, une grande valeur de d'_{ij} soutient l'hypothèse $R_{ij} = 0$, alors qu'une petite valeur de d'_{ij} ne soutient pas spécifiquement $R_{ij} = 1$ ou $R_{ij} = 0$, puisque deux objets différents peuvent avoir une vitesse similaire. Par conséquent, la forme appropriée pour la fonction de masse m^v_{ij} est la suivante :

$$m^v_{ij}(\{0\}) = \alpha' \left(1 - \psi(d'_{ij})\right) \tag{5.21a}$$

$$m^v_{ij}(\Theta_{ij}) = 1 - \alpha' \left(1 - \psi(d'_{ij})\right), \tag{5.21b}$$

où, comme ci-dessus, $\alpha' \in [0, 1]$ est le degré de confiance dans l'information du capteur, ψ est une fonction décroissante dans $[0, 1]$. Cette fonction peut être choisie de la même forme que (5.20), avec peut-être un coefficient différent γ'.

Classe

Dans plusieurs applications, les objets sont classés dans différentes catégories comme les piétons, les voitures, les motocycles, etc. Soit Ω l'ensemble de classes, et soient m_i et m_j les fonctions de masse représentant l'information sur la classe des objets e_i et f_j. Une telle fonction de masse peut être fournie, par exemple, par des classifieurs évidentiels décrits dans [Den95, Den00]. Ristic et Smets [RS07] ont exploré le problème d'association d'objets en utilisant une telle information de classe. Ils ont fait l'hypothèse que l'égalité de classe implique l'égalité des objets, une hypothèse discutable surtout quand le nombre de classes n'est pas largement supérieur au nombre d'objets. Le calcul de la fonction de masse m''_{ij} dans le cadre de discernement Θ_{ij} à partir des fonctions de masse m_i et m_j dans Ω peut être effectué comme suit.

Supposons que S_{ij} représente l'hypothèse les objets appartiennent à la même classe, et que $\Omega_{ij} = \{S_{ij}, \overline{S}_{ij}\}$, où $\overline{S}_{ij}$ est la négation de S_{ij}. La crédibilité et la plausibilité de S_{ij} calculées à partir de m_i et m_j ont les expressions suivantes [DM04] :

$$Bel(\{S_{ij}\}) = \sum_{\omega \in \Omega} m_1(\{\omega\}).m_2(\{\omega\}) = \eta_{ij}, \tag{5.22a}$$

$$Pl(\{S_{ij}\}) = 1 - \sum_{A \cap B = \emptyset} m_1(A).m_2(B) = 1 - \kappa_{ij}, \tag{5.22b}$$

où κ_{ij} est le degré de conflit (1.7) entre m_i et m_j. La fonction de masse correspon-

dante μ_{ij} dans Ω_{ij} est

$$\mu_{ij}(\{S_{ij}\}) = \eta_{ij}, \tag{5.23a}$$
$$\mu_{ij}(\{\overline{S}_{ij}\}) = \kappa_{ij}, \tag{5.23b}$$
$$\mu_{ij}(\Omega_{ij}) = 1 - \eta_{ij} - \kappa_{ij}. \tag{5.23c}$$

Il est évident que deux objets de classes différentes ne peuvent pas être identiques, alors que deux objets de la même classe peuvent être identiques ou non, ce qui peut être exprimé formellement de la manière suivante :

$$\overline{S}_{ij} \Rightarrow (R_{ij} = 0), \tag{5.24a}$$
$$S_{ij} \Rightarrow (R_{ij} = 0) \text{ ou } (R_{ij} = 1). \tag{5.24b}$$

Par conséquent, une fonction de masse m^c_{ij} dans Θ_{ij} peut être calculée à partir de μ_{ij} en transférant la masse κ_{ij} sur $\{0\}$ et le reste $1 - \kappa_{ij}$ sur Θ_{ij}. Nous obtenons ainsi :

$$m^c_{ij}(\{0\}) = \kappa_{ij}, \tag{5.25a}$$
$$m^c_{ij}(\Theta_{ij}) = 1 - \kappa_{ij}. \tag{5.25b}$$

Pour chaque paire d'objets (e_i, f_j), une fonction de masse m_{ij} sur Θ_{ij} représentant toute l'évidence disponible sur R_{ij} peut être finalement obtenue en combinant m^p_{ij}, m^v_{ij} et m^c_{ij} par la règle de Dempster :

$$m_{ij} = m^p_{ij} \oplus m^v_{ij} \oplus m^c_{ij}. \tag{5.26}$$

5.5.2. Expériences avec des données simulées

L'approche développée ci-dessus a été testée sur des données simulées. Chaque instance du problème d'association a été générée de la manière suivante. Nous considérons que chacun des deux capteurs perçoit n objets, parmi lesquels 80% sont des objets réels et 20% sont faux. La position x_i de chaque objet réel i a été générée à partir d'une distribution uniforme dans le carré $[0, 5]^2$ (cm), tandis que sa vitesse v_i a été générée avec une direction uniformément répartie dans $[0, 2\pi)$ et une norme uniformément répartie dans $[0, 0.5]$. Les objets sont supposés appartenir

à une classe parmi deux classes équiprobables (1 ou 2) ; ils sont décrits par un attribut y_i distribué normalement avec un écart type $\sigma = 2$ et une moyenne -1 dans la première classe et $+1$ dans la deuxième classe.

Pour chaque objet réel i, chaque capteur est supposé observer des versions bruitées de x_i, v_i et y_i définies comme suit :

$$\widehat{\mathbf{x}}_i = \mathbf{x} + \boldsymbol{\epsilon}_i, \quad \widehat{\mathbf{v}}_i = \mathbf{v} + \boldsymbol{\epsilon}'_i, \quad \widehat{y}_i = y_i + \epsilon''_i, \tag{5.27}$$

où $\boldsymbol{\epsilon}_i$ et $\boldsymbol{\epsilon}'_i$ sont des bruits Gaussiens de moyenne $(0, 0)$ et de variance $0.04I$ (I est la matrice identité) et ϵ''_i un bruit Gaussien de moyenne 0 et d'écart-type 0.2. Les fonctions de masse m^p_{ij} et m^v_{ij} sont calculées comme détaillé à la section 5.5.1 avec $\alpha = \alpha' = 0.9$ et $\gamma = \gamma' = 0.5$ et la distance euclidienne. Chaque capteur est supposé calculer une fonction de masse m_i sur la classe de l'objet i à partir de sa caractéristique bruitée $\widehat{y}_i$ en utilisant la formule suivante :

$$m_i(\{1\}) = \frac{f_1(\widehat{y}_i)}{f_1(\widehat{y}_i) + f_2(\widehat{y}_i)}, \quad m_i(\{2\}) = 1 - m_i(\{1\}), \tag{5.28}$$

où f_k est la densité de distribution de y_i dans la classe k.

Les faux objets ont été générés de la même façon que les objets réels mais d'une façon indépendante des deux capteurs. Un exemple d'un tel problème d'association avec $n = 40$ objets (parmi lesquels 32 sont des objets réels) est représenté sur la figure 5.1.

Le nombre n d'objets varie entre 5 et 80. Pour chaque valeur de n, 30 problèmes d'association ont été générés et résolus avec $\lambda = 0$. La qualité de l'association a été mesurée par deux critères : la précision et le rappel définis, respectivement, comme étant la proportion de couples associés qui sont corrects, et la proportion de paires d'objets réels qui ont été associés. La Figure 5.2 montre la moyenne de la précision et du rappel en fonction de n en prenant en compte en premier lieu, la position uniquement, en second lieu la position et la vitesse, puis toutes les sources d'informations (position, vitesse et classe). Comme prévu, la méthode utilise efficacement les informations supplémentaires contenues dans les fonctions de masse, et les performances se dégradent avec n. La Figure 5.3 montre la moyenne de temps de calcul en fonction de n ; l'algorithme d'association a été programmé en Matlab et exécuté sur un ordinateur personnel MacBook Pro.

Nous avons aussi étudié l'influence de λ dans (5.12) et (5.14). Nous rappelons

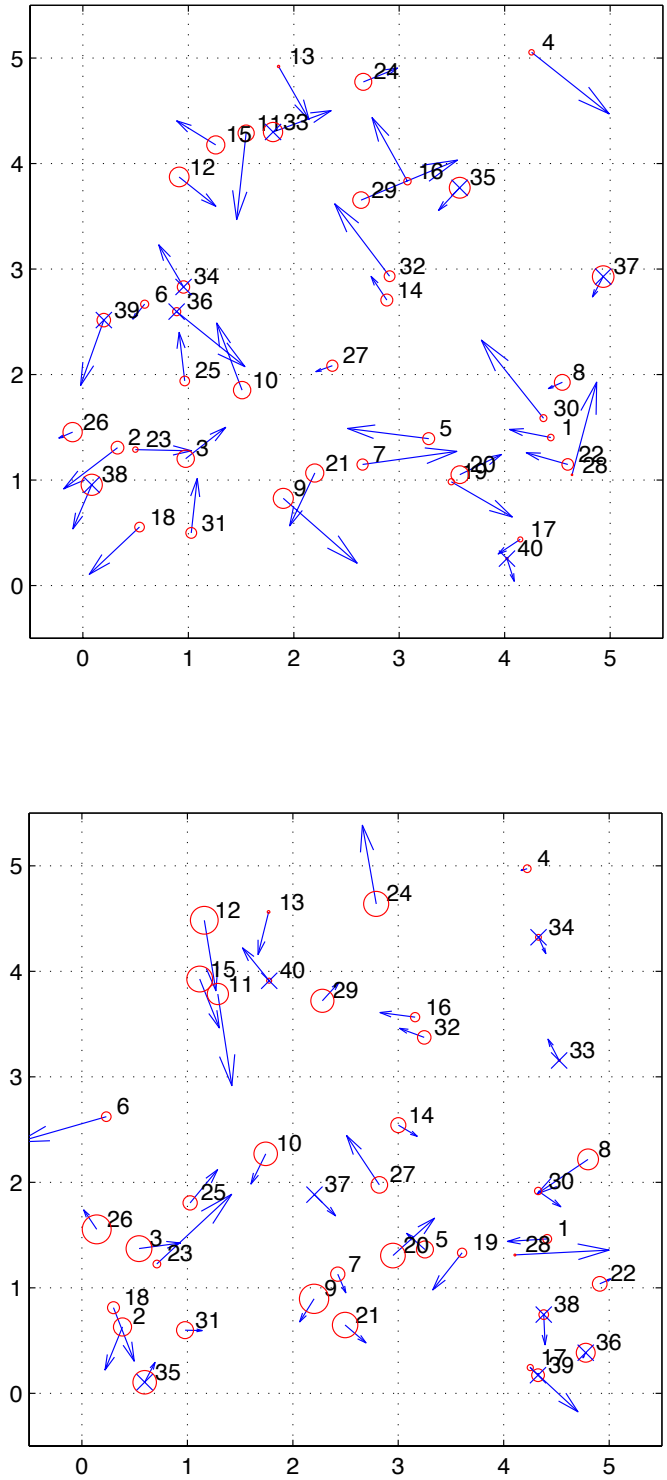


Figure 5.1.: *Exemple d'un problème d'association avec 40 objets détectés avec deux capteurs (a et b). Les objets 33 à 40 (marqués avec un ×) sont faux. La dimension du cercle est proportionnelle à la probabilité estimée de la classe 1.*

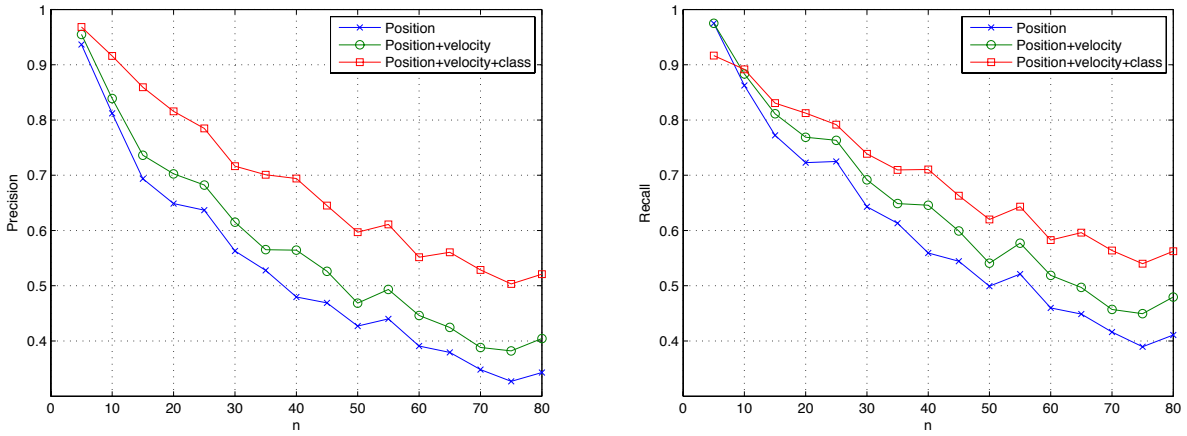


Figure 5.2.: *Moyenne de la précision (a) et du rappel (b) de la mise en correspondance en fonction du nombre d'objets n.*

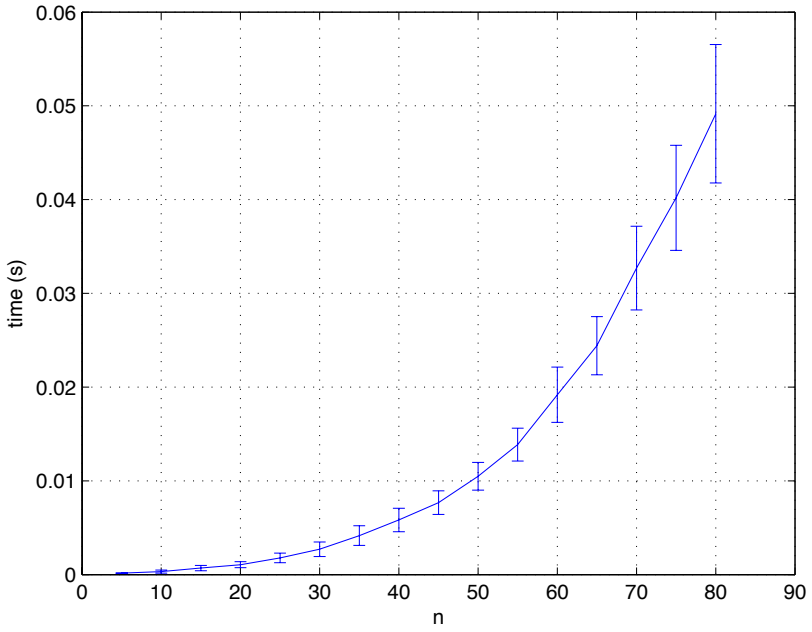


Figure 5.3.: *Moyenne du temps d'exécution (en secondes) plus ou moins l'écart type en fonction de n.*

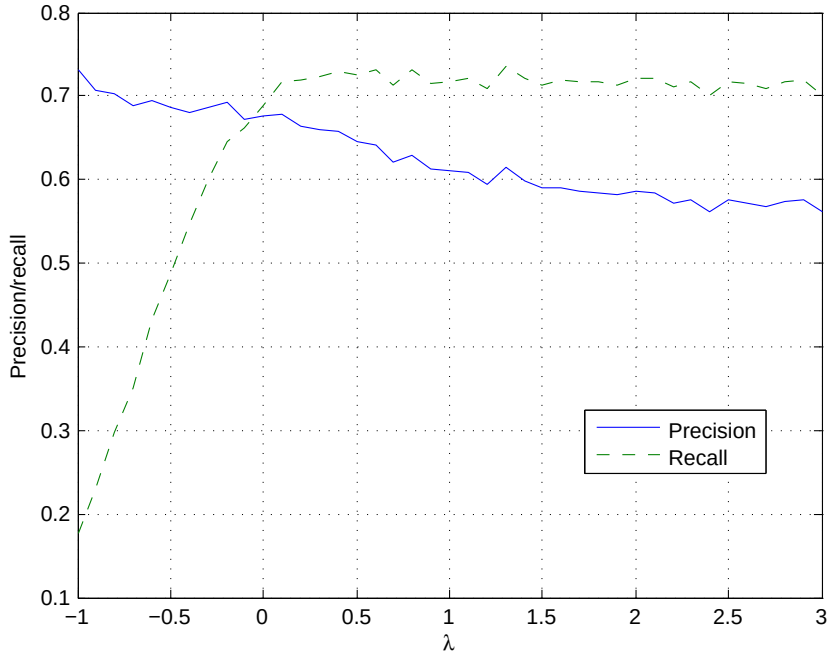


FIGURE 5.4.: *Moyenne de la précision et du rappel (moyenne de 200 problèmes d'association avec $n = 50$ objets), en fonction de λ.*

que ce paramètre permet d'introduire des connaissances a priori sur le nombre d'associations, une valeur positive (respectivement, négative) favorisant l'appariement d'un plus grand nombre (respectivement, un plus petit nombre) de paires d'objets. Par conséquent, avec l'augmentation de λ nous pouvons nous attendre à une augmentation du rappel (plus de vrais paires d'objets appariées) et à la diminution de la précision (plus de paires associées à tort). Une expérience a été réalisée avec $n = 40$ objets générés comme expliqué ci-dessus, avec λ compris entre -1 et $+3$. Pour chaque valeur de λ, 200 problèmes d'association ont été générés. Les résultats présentés sur la Figure 5.4 confirment que les différentes valeurs de λ conduisent à différents compromis entre la précision et le rappel. Cependant, une valeur élevée de la précision ne peut être obtenue qu'en faveur d'une très faible valeur de rappel pour ce problème. En attribuant différents poids à la précision et au rappel, il serait possible de trouver une valeur optimale de λ. Dans la plupart des applications, il suffit de prendre $\lambda = 0$ comme une valeur par défaut.

Finalement, notre algorithme a été comparé avec la méthode de Mercier [MLJ11]. Néanmoins, la comparaison ne peut être faite que pour une faible valeur de n

($n \leq 7$) à cause de la complexité calculatoire de cette méthode qui limite son application à un petit nombre d'objets. Pour ce problème, les deux méthodes donnent des solutions identiques dans presque tous les cas. Cependant, comme le montre la figure 5.5, le temps d'exécution de la méthode de Mercier augmente exponentiellement avec n, alors que la complexité de notre méthode est polynomiale.

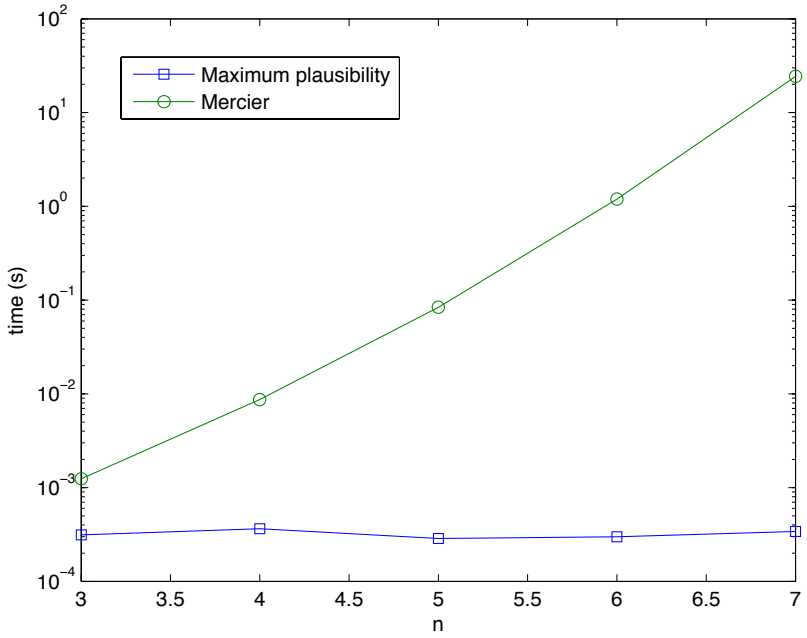


FIGURE 5.5.: *Moyenne de temps d'exécution (en secondes) pour notre méthode (maximum plausibility) et l'algorithme de Mercier, en fonction de n.*

5.5.3. Expérimentations avec des données réelles

Notre approche a été appliquée dans le cadre d'une application d'aide à la conduite [FCD08]. Un véhicule a été instrumenté par deux capteurs : Mobileye, une caméra équipée d'un système de fusion et IBEO Alasca-XT, un télémètre Laser. Chaque capteur a son système de traitement d'objets qui lui permet de détecter et de classifier les objets.

Pour chaque capteur, les fonctions de masse sur la position m_{ij}^{p} sont calculées par les équations (5.19) et (5.20) avec $\alpha = 0.9$, $\gamma = 0.1$ et d_{ij} définie comme la distance

de Mahalanobis :

$$d_{ij} = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)' \left(P_i + P_j\right)^{-1} (\mathbf{x}_i - \mathbf{x}_j)}, \tag{5.29}$$

où $\mathbf{x}_i$ est la position estimée du centre de l'objet i et P_i est la matrice de covariance estimée correspondante.

Chaque capteur prédit aussi la classe des objets, exprimée dans différents cadres de discernement. Le laser classe les objets comme piéton ou non piéton en utilisant l'algorithme développé par Fayad et al [FCD08] ; cet algorithme calcule une fonction de masse sur deux classes. La cadre de discernement du système Mobileye contient cinq classes : piéton, véhicule, camion, motocycle, vélo (un raffinement du cadre du discernement du laser). Etant donné que ce dernier ne donne pas d'indicateur de confiance dans la prédiction des classes, une masse de 0.9 est affectée à la classe prédite (piéton ou non piéton), et 0.1 pour le cadre de discernement. Pour chaque paire de fonctions de masse m_i et m_j, une fonction de masse m^c_{ij} est calculée en se basant sur leur degré de conflit selon l'équation (5.25), et est combinée avec m^p_{ij} par la règle de Dempster. L'information sur la vitesse n'est pas utilisée dans cette application.

Le jeu de données utilisé contient 58 problèmes d'appariement. Le capteur Mobileye a détecté entre un et trois objets avec une moyenne de deux objets alors que le télémètre laser a été moins sélectif et a détecté entre deux et 23 objets (moyenne : 8.5). Un exemple de problème d'association est illustré sur les figures 5.6a et 5.6b, et le résultat correspondant est présenté sur la figure 5.6c. Les fonctions de masse m^p_{ij} sur la position et m^c_{ij} sur la classe sont représentées dans les tableaux 5.1 et 5.2. Le tableau 5.3 présente la fonction de contour correspondante. La relation la plus plausible R^* dans ce cas est :

$$R^* = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{5.30}$$

Les moyennes de la précision et du rappel de notre algorithme (avec $\lambda = 0$) sont 0.75 et 0.90, respectivement (avec des écarts type 0.33 et 0.30). L'algorithme de Mercier donne des résultats identiques, avec un temps de calcul beaucoup plus grand (119 secondes en moyenne, contre 0.23 pour notre méthode).

Table 5.1.: *Masse sur la position m^p_{ij} où $m^p_{ij}(\{1\})$ représente le fait que les objets sont associés et $m^p_{ij}(\{0\})$ la non association*

$m^p_{ij}(\{1\})$	f_1	f_2	f_3	f_4
e_1	0.45	0.01	0.32	0.68
e_2	0.71	0.02	0.34	0.39
e_3	0.01	0.73	0.02	0.01
$m^p_{ij}(\{0\})$	f_1	f_2	f_3	f_4
e_1	0.45	0.89	0.58	0.22
e_2	0.18	0.88	0.56	0.51
e_3	0.9	0.17	0.88	0.89

Table 5.2.: *Masse sur la classe m^c_{ij}*

$m^c_{ij}(\{0\})$	f_1	f_2	f_3	f_4
e_1	0	0.77	0	0
e_2	0	0.77	0	0
e_3	0.5	0	0.57	0.76
$m^c_{ij}(\Theta_{ij})$	f_1	f_2	f_3	f_4
e_1	1	0.23	1	1
e_2	1	0.23	1	1
e_3	0.5	1	0.43	0.24

Table 5.3.: *Fonctions de contour pl_{ij}. Les deux nombres de chaque case sont $pl_{ij}(1)$ et $pl_{ij}(0)$.*

pl_{ij}	f_1	f_2	f_3	f_4
e_1	0.55	0.02	0.41	0.78
	0.55	0.99	0.68	0.31
e_2	0.81	0.03	0.43	0.49
	0.28	0.99	0.66	0.6
e_3	0.05	0.82	0.05	0.02
	0.99	0.27	0.99	0.99

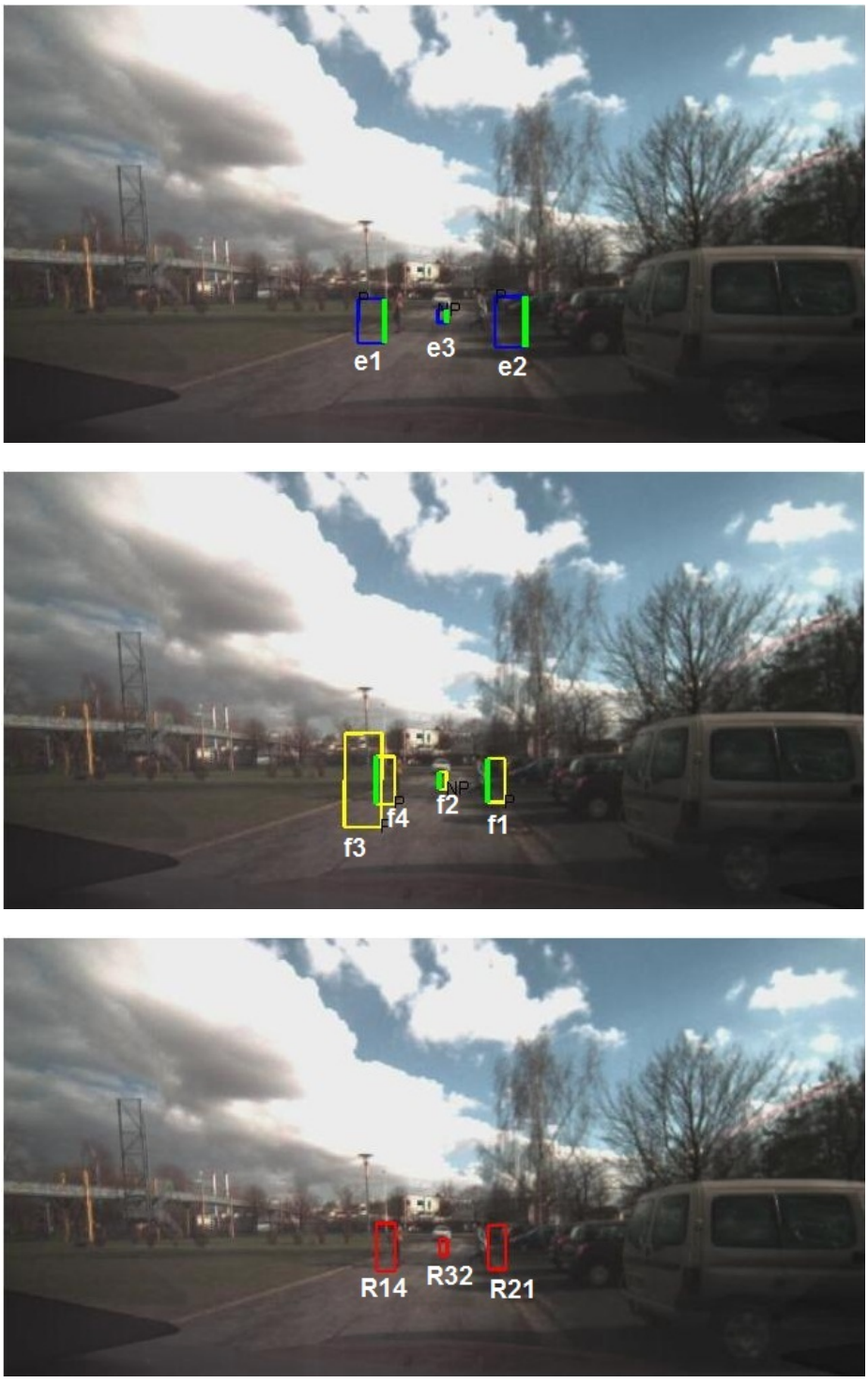


Figure 5.6.: *Exemple d'un problème d'association : le télémètre laser détecte trois objets (a) alors que le capteur Mobileye détecte quatre objets y compris une fausse alarme (b). L'algorithme d'association associe correctement les trois objets réels (c).*

5.5.4. Eléments d'évidence conflictuels

Afin d'étudier le comportement de la méthode en cas de conflit entre les éléments d'évidence, nous avons testé l'algorithme d'association sur des données simulées créées par A. Houenou [HBCB12]. Un véhicule V_0 est équipé de deux capteurs : S_1 (champ de vision : *portée maximale* $= 60m$ et $\alpha = 45°$) et S_2 (champ de vision : *portée maximale* $= 100m$ et $\alpha = 120°$). Il peut détecter d'autres véhicules sur une autoroute (jusqu'à neuf véhicules) comme représenté dans la figure 5.7. Chaque véhicule détecté par V_0 est caractérisé par ses informations cinématiques (position et vitesse), sa matrice de covariance sur la position et sur la vitesse et sa classe. Nous associons tous les objets détectés par S_1 et S_2 à chaque pas de temps. Dans cette simulation, nous ne prenons pas en compte les occultations afin de détecter la plupart des objets sur la route.

Pour évaluer le conflit nous calculons la plausibilité de la relation la plus plausible R^* en utilisant l'équation 5.9. Dans le cas des évidences conflictuelles, la plausibilité a une faible valeur. Nous avons testé un scénario d'une durée de 25s (avec un pas de temps de 0.01s) et nous avons simulé différentes situations de conflits. Les valeurs maximales de plausibilité sont présentées sur la figure 5.8. La plausibilité est faible entre [5s,18s] ; elle est égale à 1 dans le cas où un capteur ne détecte pas les objets dans son champ de vision, auquel cas il n'y a pas d'ambiguïté dans le processus d'association.

La figure 5.9 montre la situation à $t = 4.8s$, où la plausibilité est égale à 0.18. Le capteur S_2 de V_0 détecte $\{V_1, V_5\}$ pendant que S_1 détecte $\{V_1\}$. V_1 est associé à V_1.

Dans la figure 5.10, S_2 détecte sept objets à l'instant $t = 17.3s$: $\{V_1, V_2, V_3, V_4, V_5, V_6, V_7\}$ alors que S_1 détecte, au même instant, trois objets : $\{V_1, V_2, V_7\}$. La relation la plus plausible R^* dans ce cas est :

$$R^* = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5.31}$$

et pl est égale à 0.013. Cette faible valeur indique un degré élevé de conflit parce que les autres véhicules sont des candidats plausibles pour l'association. Cependant, cet exemple montre que l'algorithme d'association peut trouver la solution la plus plausible même en cas de conflit.

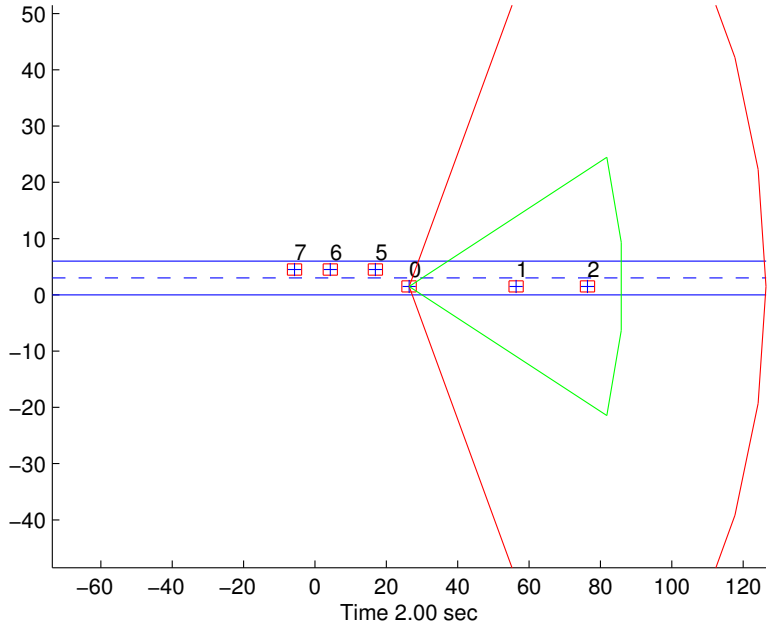


FIGURE 5.7.: *Exemple de données simulées : le véhicule V_0 est équipé de S_1 (en vert) et S_2 (en rouge)*

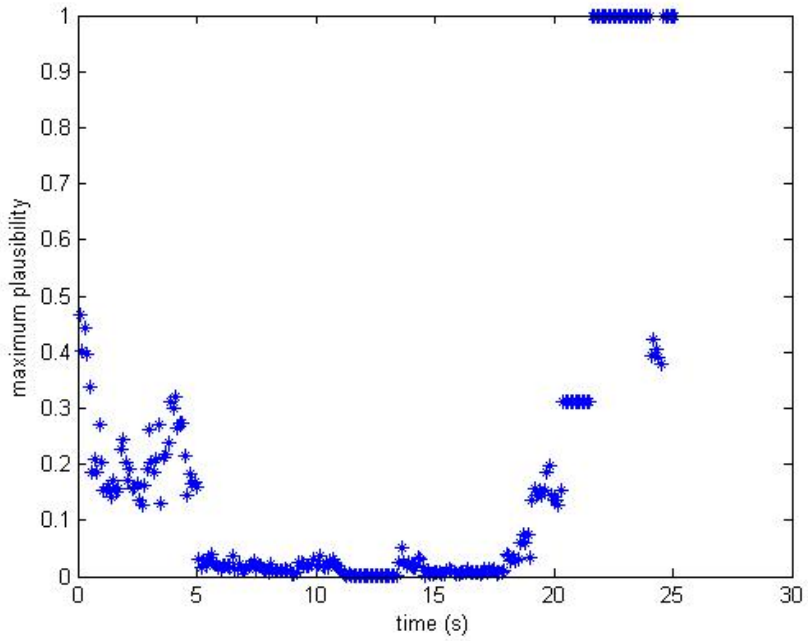


FIGURE 5.8.: *Plausibilité de R^**

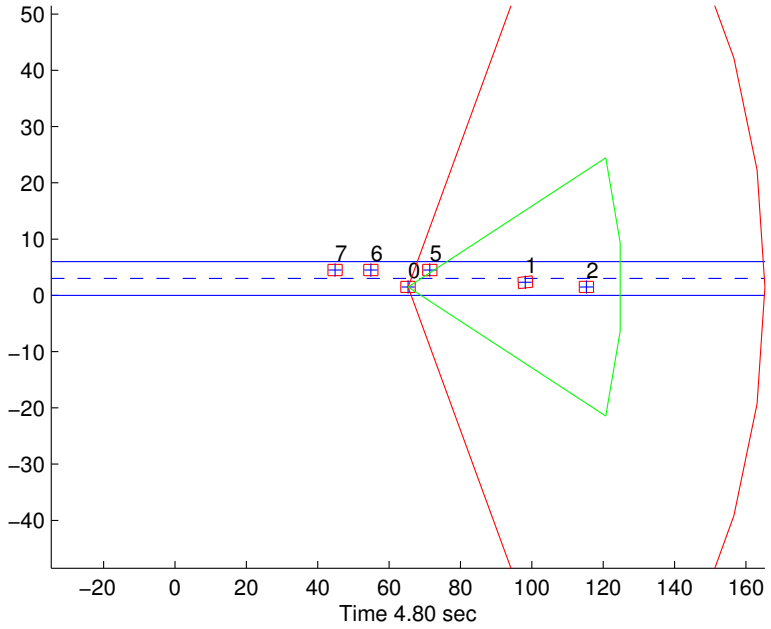


FIGURE 5.9.: *Scénario à l'instant t=4.8s : le véhicule V_0 est équipé de S_1 (en vert) et S_2 (en rouge)*

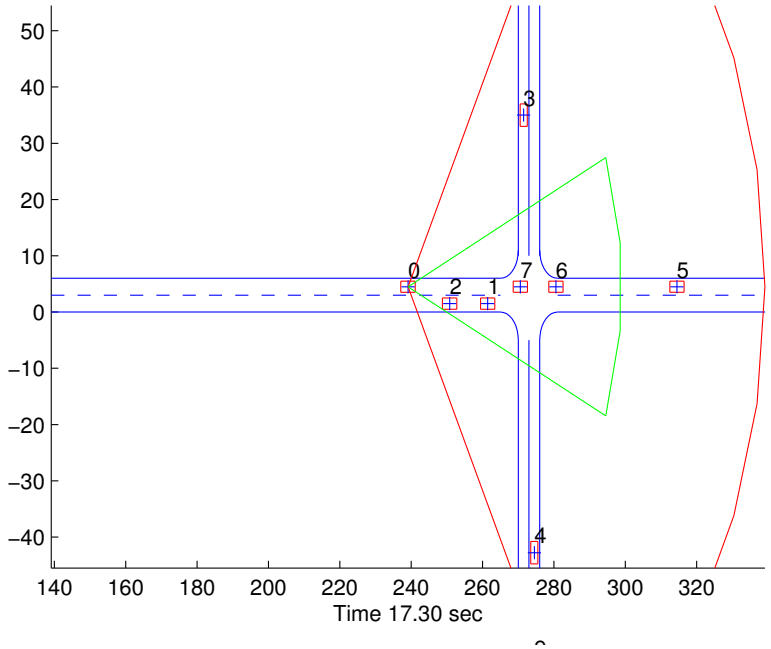


FIGURE 5.10.: *Scénario à l'instant t=17.3s : le véhicule V_0 est équipé de S_1 (en vert) et S_2 (en rouge)*

5.6. Conclusion

L'association d'objets est un problème très important dans beaucoup d'applications et un élément clé dans de nombreux systèmes de fusion de données. Les fonctions de croyance ont souvent été considérées comme un formalisme idéal pour représenter et combiner l'information dans les applications multi-capteurs (voir, par exemple, [MLJ11, DOO13, RS07, Sch08]). Cependant, les méthodes développées jusqu'à présent pour résoudre le problème d'affectation dans le cadre de Dempster-Shafer ont été empiriques et souvent très coûteuses en temps (voir, par exemple [RC97, GRLD03, MACC05, MLJ11, DOO13]).

Dans ce chapitre, l'information sur l'association possible pour n'importe quelle paire d'objets a été modélisée par des fonctions de masse de Dempster-Shafer définies dans le cadre de toutes les relations possibles et pertinentes entre deux ensembles d'objets. La plausibilité d'une relation, après la mise en commun de toutes les fonctions de masse disponibles, peut être calculée d'une manière efficace et optimisée pour trouver la relation la plus plausible. Il a été démontré que ce problème est équivalent à un problème d'affectation linéaire, qui peut être résolu en temps polynomial en utilisant, par exemple, l'algorithme hongrois. Cette solution est à la fois optimale et beaucoup plus efficace que d'autres approches développées dans le cadre des fonctions de croyance [MLJ11].

Le problème d'association tridimensionnelle, une extension à ce travail a été aussi exploré (Annexe C). Ce problème se pose lorsqu'on fusionne les données fournies par trois capteurs. Il peut être formalisé de la même manière que le problème à deux dimensions, qui est $\mathcal{NP}$-difficile, ce qui rend peu probable qu'une solution puisse être trouvée en temps polynomial. Le développement de méthodes heuristiques pour trouver des solutions approchées à ce problème est une perspective intéressante pour des recherches ultérieures.

CHAPITRE 6

Conclusions et perspectives

6.1. Conclusion

Ce travail de recherche a été consacré à la fusion de données dans un réseau de véhicules et, en particulier, à la gestion de la confiance dans un tel réseau. Nous nous sommes intéressés à la représentation et à la gestion des incertitudes afin d'estimer, d'une part, la confiance dans les noeuds du réseau et, d'autre part, la confiance dans les données échangées. Ainsi, nous avons proposé et développé un algorithme de fusion de données distribuée utilisant le formalisme des fonctions de croyance. L'algorithme de fusion distribuée se base sur le principe que chaque noeud possède deux connaissances : une connaissance locale et une connaissance publique (distribuée). La connaissance locale est ce que le véhicule (le noeud) détecte lui-même dans son environnement. La connaissance publique est le résultat de la combinaison avec ce qui provient des autres noeuds du réseau. C'est la connaissance publique qui est rediffusée dans le réseau. Ces connaissances peuvent être incertaines. Grâce à l'utilisation d'opérateurs de combinaison adaptés à la provenance des informations, on profite de la redondance des informations pour diminuer l'incertitude en évitant par ailleurs le phénomène de "data incest" induit par les cycles. L'affaiblissement des fonctions de masse, la combinaison de Dempster et la règle prudente idempotente permettent d'implémenter un opérateur de fusion distribuée garantissant l'auto-stabilisation.

Afin de valider l'algorithme de fusion distribuée, nous avons développé deux applications : la détection d'attaque sybil et la carte dynamique distribuée. La détection d'attaque sybil se base sur la quantification de la confiance dans les noeuds du

réseau pour détecter les faux noeuds créés par un noeud malveillant. La connaissance locale est établie à partir de l'analyse des puissances reçues et des positions annoncées. La connaissance distribuée est la croyance dans le fait que les noeuds du réseau sont vrais ou faux. Cette méthode a été validée par simulation sur un réseau contenant un véhicule malveillant créant différents noeuds sybils. L'algorithme converge vers la solution en un temps acceptable par rapport à l'application.

L'application de la carte dynamique distribuée se base sur le principe de la perception coopérative où les véhicules coopèrent pour étendre leur champ de perception. Chaque véhicule est équipé de capteurs lui permettant de détecter les objets dans son environnement proche et de construire sa connaissance locale. L'algorithme de fusion distribuée permet de construire une carte de l'environnement dynamique comprenant les objets dans le champ de vision du capteur ainsi que ceux envoyés par les autres véhicules. Il en résulte une carte dynamique distribuée offrant une perception augmentée de l'environnement. La mise en oeuvre d'une telle application est complexe et a nécessité de nombreux traitements : recalage temporel et spatial, mise en correspondance des objets par un algorithme d'association et simulation des messages et des échanges dans le réseau. L'intérêt de la fusion de cartes dynamiques pour la perception augmentée a été validé par simulation sur différents scénarios de situation routière impliquant plusieurs véhicules.

Comme dans tout algorithme de fusion, l'association est une étape importante qui permet de mettre en correspondance les données avant de les fusionner. Nous avons développé une nouvelle méthode d'association d'objets dans le cadre des fonctions de croyance. Les informations d'attributs fournies par les capteurs sont transformées en fonctions de masse puis combinées par la règle de Dempster. Cette méthode permet de trouver la relation la plus plausible entre deux ensembles d'objets. Le problème d'association optimale est équivalent à un problème d'affectation linéaire. La méthode a été validée par des expérimentations sur des données réelles et des données simulées et a montré de très bonnes performances en temps d'exécution par rapport à la méthode de référence. Cette approche a été utilisée dans l'application de la carte locale dynamique distribuée. L'extension tridimensionnelle de ce problème (avec trois ensembles d'objets), également formalisée, s'avère NP-difficile.

6.2. Perspectives

Maintenant que nous avons montré, sur des simulations, l'intérêt de l'algorithme de fusion distribuée, il est envisagé de profiter des plateformes du laboratoire HEUDIASYC pour réaliser des expérimentations en situations réelles. Les plateformes Airplug [air] développée par Bertrand Ducourthial et Pacpus [pac] (plusieurs véhicules équipés) permettraient dans un premier temps d'enregistrer et de rejouer les données de plusieurs véhicules puis de tester nos algorithmes en conditions réelles. Ce travail nécessite de porter les programmes sur ces plateformes et de définir des scénarios adaptés. Des expérimentations avec trois véhicules ont été menées en mai 2013 mais des problèmes sur les acquisitions des données GPS nous ont conduits à tester nos algorithmes sur des données simulées. Il est prévu dans le projet COMOSEF [com] (projet Celtic 2012-2015) d'appliquer cet algorithme pour des applications météo dans un réseau de véhicules. D'autres domaines d'applications sont possibles comme la perception coopérative dans un réseau de robots mobiles tel que cela a été envisagé dans les deux stages que j'ai co-encadrés. Le premier a été effectué par Hassan Kanj dans le cadre d'un master et le second par Saad Mohammad dans le cadre d'un projet de fin d'études pour un diplôme d'ingénieur.

Une autre perspective est d'étudier l'influence de la perte des données sur la fusion distribuée. L'échange de données par wifi peut s'accompagner de perte de données pour différentes raisons, comme la présence d'obstacles (bâtiments), les conditions environnementales, etc. Les pertes de données peuvent être de 25 à 30% et dans certains cas les véhicules peuvent ne recevoir que 10% du message envoyé. Il est donc important d'étudier le cas où les véhicules ne reçoivent pas les messages et l'influence de ce problème sur la fusion distribuée. Cette étude peut être menée sur les expérimentations réelles dans lesquelles les véhicules peuvent communiquer ou en utilisant l'émulateur Airplug.

Enfin, pour aller plus loin, nous pouvons envisager de gérer en même temps la confiance dans les noeuds et dans les données échangées dans le réseau. Pour la carte dynamique distribuée, nous avons considéré que les noeuds sont fiables. Mais dans certains cas on peut avoir un noeud malveillant qui peut modifier les fonctionnalités du VANet. Il serait intéressant d'étudier la gestion des deux types de confiance. On pourrait en particulier évaluer grâce à la mesure du conflit le désaccord entre plusieurs sources de données et utiliser cette information pour définir la confiance dans les noeuds.

Annexes

ANNEXE A

Sécurité dans les VANets

A.1. Réseau de Véhicule

Un réseau ad hoc de véhicules ou VANet est constitué de véhicules capables de s'échanger des informations par voie radio dans le but d'améliorer la sécurité routière ou de permettre l'accès à internet pour les passagers. Par rapport à un réseau ad hoc classique, le VANet se différencie par une forte mobilité des nœuds rendant la topologie du réseau fortement dynamique. Les réseaux mobiles actuels peinent à satisfaire des communications haut débit à grande vitesse. La mise en place de réseaux de communication inter-véhicules ne fera pas double emploi avec les réseaux mobile d'opérateurs.

Dans ces réseaux de véhicules, les services proposés permettent de distinguer plusieurs communications possibles : Communication de véhicule à véhicule, de véhicule à infrastructure et communications hybrides. Pour les communications de véhicule à véhicule, les applications sont basées sur la simple communication inter-véhicule et n'impliquent pas d'infrastructure. Par contre les communications de véhicule à infrastructure utilisent des points d'infrastructure (road side units ou RSUs). La communication hybride est la combinaison de ces deux types de communications, elle permet d'obtenir une communication très intéressante dans le cas où les portées des infrastructures sont limitées, les véhicules sont utilisés pour étendre cette portée.

Voici quelques exemples de situations où la communication inter-véhicules peut augmenter la sécurité routière :

– Alerte en cas d'accidents : Les véhicules se dirigeant vers le lieu de l'accident sont avertis des conditions de circulation modifiées. Cette information est retransmise aux véhicules arrivant dans la zone concernée.
– Alerte en cas de ralentissement anormal (bouchon, travaux, intempérie, etc.) : Le message d'alerte est émis par un véhicule détectant les difficultés de circulation (freinage important par exemple, déclenchement des feux de détresse, pluie). Comme pour le message d'alerte d'accident, ce message doit être transmis aux autres véhicules de façon efficace et rapide.

A.2. Sécurité dans les VANets

Comme dans tout réseau, un VANet peut être sujet à différentes attaques. Il est donc impératif de comprendre ces attaques afin de sécuriser le réseau.

Les adversaires : Les attaquants sont divisés en plusieurs catégories, on va citer les plus connus :

– Conducteurs égoïstes : La majorité des pilotes dans un réseau est honnête mais ça n'empêche pas l'existence de quelques conducteurs qui veulent profiter des avantages spécifiques du système. Dans une telle situation, les conducteurs peuvent envoyer des fausses informations pour détourner le trafic et profiter d'une route sans circulation.
– Les Eavesdroppers (oreilles indiscrètes) : Ces adversaires cherchent à recueillir des informations sur les conducteurs et les utiliser pour comprendre leur comportement et le motif de la circulation.
– Les pirates : Ces adversaires essaient de pirater n'importe quel système déployé publiquement. Leur but est de trouver des 'bugs' dans le logiciel et provoquer des perturbations de circulation juste pour le plaisir.
– Les initiés : Ces adversaires comprennent les personnes travaillant dans les entreprises de voiture et l'installation du système de communication inter-véhicule. Ils sont capables de charger des logiciels malveillants dans les voitures qui pourraient causer d'immenses dégâts.
– Les attaquants malveillants : Ils pourraient être des criminels ou des terroristes ayant accès à des outils plus sophistiqués que des attaquants normaux. Les criminels peuvent avoir des cibles spécifiques et peuvent perturber la circulation des véhicules. Ce sont les plus dangereux des attaquants et des mesures spéci-

fiques doivent être prises pour protéger le système contre telles attaques.

Les attaques : Les VANets sont sensibles à divers types d'attaques. Elles varient en fonction de la situation, l'intention de l'attaquant, etc. Nous donnons un bref aperçu des grandes attaques possibles.

- "Denial of service" : Cela peut être fait par le blocage de l'accès à un canal de communication par transmission de puissance élevée ou par injection de messages fictifs. Le blocage peut se faire facilement et perturber un réseau de communication. En outre, un attaquant pourrait injecter un grand nombre de messages fictifs dans le réseau pour inonder le réseau et ne pas laisser des messages de sécurité atteindre les destinataires souhaités.
- Usurpation d'identité : Un attaquant peut prendre l'identité de quelqu'un d'autre et gagner certains avantages. Par exemple, il peut usurper l'identité d'un véhicule de secours pour accéder à l'autoroute ou il peut envoyer l'identité d'une autre personne pendant un accident pour échapper à la responsabilité.
- Falsification du message : Un attaquant peut envoyer de faux messages dans un réseau VANet tels que les avertissements de faux danger pour détourner le trafic d'une route.
- Altération des messages : Cette forme d'attaque se fait soit en changeant une partie d'un message ou bien son contenu complet. Par exemple, un avertissement de danger peut être changé et peut causer des accidents.
- Retard et suppression des messages : En cas d'accident, certains véhicules peuvent retarder ou supprimer les messages de sécurité. En supprimant les messages de façon sélective ou en retardant leur transmission, des informations critiques peuvent ne pas être reçues par les véhicules à temps.
- Violation de la confidentialité : Pour éviter l'usurpation d'identité, une authentification des messages est nécessaire. Il faut associer l'identité des véhicules aux messages qu'ils envoient en utilisant la cryptographie à clé asymétrique. Toutefois, cela est un avantage pour les gens capables d'identifier l'expéditeur du message. Ainsi, les véhicules peuvent être suivis et n'importe qui peut identifier le propriétaire d'un véhicule. Cela peut causer différents problèmes.
- "Replay attacks" : Un véhicule peut facilement écouter et enregistrer les messages des autres véhicules et les rejouer plus tard pour avoir accès à des ressources spécifiques, pour envoyer par exemple de fausses alertes.
- Manipulation du matériel : Les pirates peuvent aussi manipuler le matériel de

sécurité d'un véhicule, voler les identités et extraire les clés cryptographiques.
- Manipulation des capteurs : Une autre attaque peut se faire au niveau des capteurs du véhicule en modifiant les informations fournies par ceux-ci, par exemple, le GPS ou les capteurs de température.
- Fausses informations : Les attaquants peuvent envoyer de fausses informations dans le réseau, cela influence le comportement des autres conducteurs.
- Attaque "Wormhole" : Il est difficile de détecter et de prévenir cette attaque. Un nœud malveillant peut enregistrer les paquets à un endroit dans le réseau et les envoyer à un autre endroit via un réseau privé partagé avec les nœuds malveillants. La gravité de l'attaque augmente si le nœud malveillant envoie uniquement des messages de contrôle à travers le tunnel et non pas les paquets de données.
- Attaque "Sybil" : Dans cette attaque, un nœud attaquant envoie des messages avec des identités multiples pour les autres nœuds du réseau. L'attaquant simule plusieurs nœuds dans le réseau. Le nœud qui usurpe l'identité des autres nœuds est appelé nœud malveillant, et les nœuds dont l'identité est usurpée sont appelés nœuds Sybil.

ANNEXE B

Carte dynamique locale

La carte dynamique locale peut être construite de différentes façons selon le type et le nombre de capteurs utilisés. Dans cette partie, nous allons détailler le cas où le véhicule est muni de différentes sources d'informations S_k et le cas où le capteur fournit des observations, c'est à dire des objets et non de pistes.

B.1. Cas 1 : Sources Multiples

Dans le cas de sources multiples, différents traitements sont à prévoir. Chaque source S_k va détecter les objets et construire sa connaissance locale notée $CDL_{j,S_k}(t) = \{O_i, m(O_i)\}^t_{j,S_k}$. La figure B.1 montre la connaissance locale dans le cas de deux sources. Dans un premier temps, nous associons les objets en utilisant l'algorithme détaillé dans le chapitre 5. S'ils sont associés, nous procédons de la manière suivante :

– fusion statique des paramètres de position et vitesse :

$$\begin{aligned} X &= P_1.(P_1+P_2)^{-1}.X_2 + P_2.(P_1+P_2)^{-1}.X_1, \\ P &= P_1.(P_1+P_2)^{-1}.P_2, \end{aligned} \tag{B.1}$$

où X représente l'état : $X = [x, y, v_x, v_y]$ et P est la matrice de covariance correspondante.

– Combinaison avec la règle de Dempster des masses sur la classe :

$$m_{c_i} = m_{c_i S_1} \oplus m_{c_i S_2}, \tag{B.2}$$

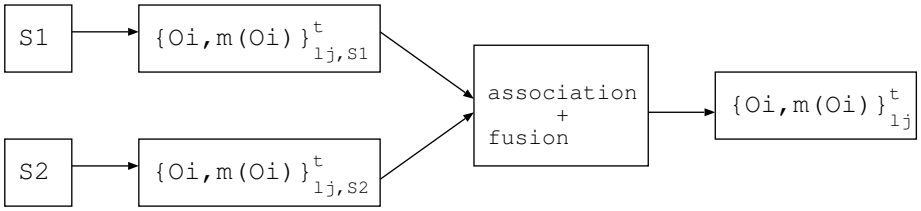


FIGURE *B.1.: Connaissance locale dans le cas de sources multiples*

– Combinaison conjonctive des masses sur l'existence de l'objet :

$$m_{O_i} = m_{O_{i_{S_1}}} \oplus m_{O_{i_{S_2}}}. \tag{B.3}$$

Si les objets ne sont pas associés, nous les gardons dans la CDL mais la masse sur l'existence n'est pas renforcée.

B.2. Cas 2 : Observations

La figure B.2 représente la mise à jour de la connaissance locale dans le cas où les informations sont des observations et non des pistes. Ceci est fait par une association de la carte à l'instant $t-1$ avec celle détectée à l'instant t. Différents cas sont à prendre en considération :

– Si les objets sont associés, nous remplaçons $CDL_j(t-1)$ par $CDL_j(t)$. Les objets O_i^{t-1} sont remplacés par O_i^t et la masse sur l'existence est le résultat de la combinaison par la règle prudente.

$$m_{O_i}^t = m_{O_i}^t \textcircled{\scriptsize /} m_{O_i}^{t-1}. \tag{B.4}$$

– Si O_i^{t-1} n'est pas associé suite à une occultation ou à une perte de l'objet, nous gardons une prédiction de l'objet et nous affaiblissons la masse : $\{\hat{\mathrm{O}}_i, {}^{\alpha}\, m(O_i\}_j^t\}$.

$$\begin{aligned} x(t+1) &= x(t) + v_x(t) * T, \\ y(t+1) &= y(t) + v_y(t) * T, \\ P(t+1) &= F.P(t).F^T + Q. \end{aligned} \tag{B.5}$$

$$\begin{aligned} {}^{\alpha}\mathrm{m(A)} &= \alpha.\mathrm{m(A)} \quad \text{avec A=}\{O_i,NO_i\}, \\ {}^{\alpha}\mathrm{m}(\Omega) &= (1\text{-}\alpha)+\alpha.\mathrm{m}(\Omega). \end{aligned} \tag{B.6}$$

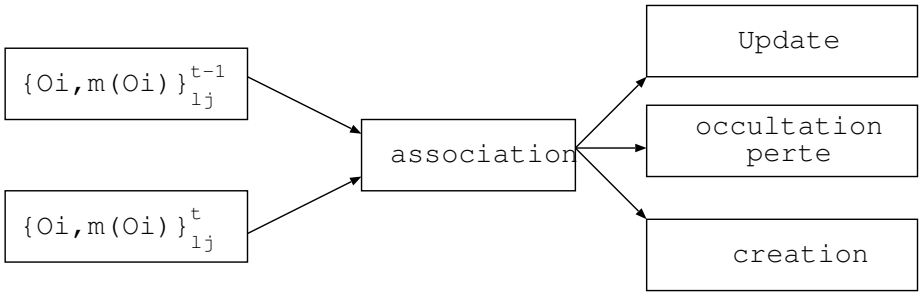


FIGURE *B.2.: Connaissance locale dans le cas des observations*

– Si O_i^t n'est pas associé, nous créons un nouvel objet $\{O_i, m(O_i\}_j^t\}$.

ANNEXE C

Extension tridimensionnelle du problème d'association

Ce travail a été réalisé avec Antoine Jouglet pour la rédaction de l'article soumis à IEEE Transactions on Cybernetics.

C.1. L'extension tridimensionnelle

Nous allons considérer dans cette partie la situation où on a plus de deux ensembles d'objets. Pour simplification de notations, et sans pertes de généralités, nous allons traiter le problème d'association en trois dimensions, dans lequel nous avons trois ensembles d'objets. Ce problème va être formalisé dans la section C.1.1 et une étude de sa complexité va être explorée dans la section C.1.2.

C.1.1. Formalisation

Soient $E = \{e_1, \ldots, e_n\}$, $F = \{f_1, \ldots, f_p\}$ et $G = \{g_1, \ldots, g_q\}$ les trois ensembles d'objets détectés par exemple par trois capteurs. Dans ce cas, nous cherchons trois relations $R \subset E \times F$, $S \subset F \times G$ et $T \subset E \times G$ représentant, respectivement, la correspondance entre les objets des ensembles E et F, F et G, E et G. Chacune de ces trois relations doit vérifier les propriétés (5.1a)-(5.1b). Soient $\mathcal{R}$, $\mathcal{S}$ et $\mathcal{T}$ les ensembles de relation R, S et T vérifiant ces propriétés. Ces relations R, S et T ne peuvent pas être déterminées de façon indépendante : notamment, si $(e_i, f_j) \in R$ et

$(f_j, g_k) \in S$, alors nous devons avoir $(e_i, g_k) \in T$. Plus généralement, pour chaque triplet d'objets $(e, f, g) \in E \times F \times G$, si deux paires d'objets sont en relation alors la troisième paire doit l'être aussi. Alors, les implications suivantes doivent être valables pour tout triplet (i, j, k) :

$$(e_i, f_j) \in R \text{ et } (f_j, g_k) \in S \quad \Rightarrow \quad (e_i, g_k) \in T \tag{C.1a}$$
$$(e_i, f_j) \in R \text{ et } (e_i, g_k) \in T \quad \Rightarrow \quad (f_j, g_k) \in S \tag{C.1b}$$
$$(f_j, g_k) \in S \text{ et } (e_i, g_k) \in T \quad \Rightarrow \quad (e_i, f_j) \in R. \tag{C.1c}$$

L'ensemble des solutions pour le problème d'association est donc l'ensemble $\mathcal{U}$ des triplets $(R, S, T) \in \mathcal{R} \times \mathcal{S} \times \mathcal{T}$ vérifiant (C.1).

Comme précédemment, nous admettons que nous recevons les éléments d'évidence de l'association pour chaque paire d'objets dans $E \times F$, $F \times G$ et $E \times G$, et ces éléments d'évidence sont transformés en forme de fonctions de masse $m_{ij\cdot}$, $m_{\cdot jk}$ et $m_{i\cdot k}$, pour tout i, j et k. Ces fonctions de masse doivent être exprimées dans un cadre de discernement commun $\mathcal{U}$ avant d'être combinées par la règle de Dempster.

Soit $U_{ij\cdot}$ l'ensemble des triplets $(R, S, T) \in \mathcal{U}$ tel que $R_{ij} = 1$, et soit $m_{ij\cdot}(\{1\}) = \alpha_{ij\cdot}$ et $m_{ij\cdot}(\{0\}) = \beta_{ij\cdot}$. Afin d'exprimer $m_{ij\cdot}$ dans $\mathcal{U}$, nous avons besoin de transférer la masse $\alpha_{ij\cdot}$ dans $U_{ij\cdot}$, $\beta_{ij\cdot}$ dans $\overline{U_{ij\cdot}}$ et $1 - \alpha_{ij\cdot} - \beta_{ij\cdot}$ dans $\mathcal{U}$. La fonction de contour correspondante dans $\mathcal{U}$ est définie dans ce qui suit :

$$pl_{ij\cdot}(R, S, T) = \begin{cases} 1 - \beta_{ij\cdot} & \text{if } R_{ij} = 1, \\ 1 - \alpha_{ij\cdot} & \text{otherwise,} \end{cases} \tag{C.2}$$

pour tout $(R, S, T) \in \mathcal{U}$, qui peut être exprimée brièvement dans :

$$pl_{ij\cdot}(R, S, T) = (1 - \beta_{ij\cdot})^{R_{ij}} (1 - \alpha_{ij\cdot})^{1 - R_{ij}}. \tag{C.3}$$

D'une façon similaire, avec des notations évidentes, les fonctions de masses $m_{\cdot jk}$ and $m_{i\cdot k}$ donnent les fonctions de contour dans $\mathcal{U}$:

$$pl_{\cdot jk}(R, S, T) = (1 - \beta_{\cdot jk})^{S_{jk}} (1 - \alpha_{\cdot jk})^{1 - S_{jk}} \tag{C.4a}$$
$$pl_{i\cdot k}(R, S, T) = (1 - \beta_{i\cdot k})^{T_{ik}} (1 - \alpha_{i\cdot k})^{1 - T_{ik}}. \tag{C.4b}$$

La combinaison des $np + pq + nq$ éléments d'évidence par la règle de Dempster

donne lieu à la fonction de contour suivante :

$$pl(R,S,T) \propto \prod_{i,j,k}(1-\beta_{ij.})^{R_{ij}}(1-\alpha_{ij.})^{1-R_{ij}}(1-\beta_{.jk})^{S_{jk}}(1-\alpha_{.jk})^{1-S_{jk}} (1-\beta_{i\cdot k})^{T_{ik}}(1-\alpha_{i\cdot k})^{1-T_{ik}}. \quad \text{(C.5)}$$

En calculant le logarithme, comme précédemment, et en considérant que α et β sont strictement inférieurs à un, nous obtenons :

$$\ln pl(R,S,T) = \sum_{i,j} w_{ij.}R_{ij} + \sum_{j,k} w_{.jk}S_{jk} + \sum_{i,k} w_{i\cdot k}T_{ik} + C, \quad \text{(C.6)}$$

où C est une constante et

$$w_{ij.} = \ln\frac{1-\beta_{ij.}}{1-\alpha_{ij.}}, \quad w_{.jk} = \ln\frac{1-\beta_{.jk}}{1-\alpha_{.jk}}, \quad w_{i\cdot k} = \ln\frac{1-\beta_{i\cdot k}}{1-\alpha_{i\cdot k}}. \quad \text{(C.7)}$$

L'association la plus plausible (R^*, S^*, T^*) peut être trouvée en résolvant le problème de programmation linéaire en nombres entiers :

$$\max_{R,S,T} \sum_{i,j} w_{ij.}R_{ij} + \sum_{j,k} w_{.jk}S_{jk} + \sum_{i,k} w_{i\cdot k}T_{ik} \quad \text{(C.8)}$$

sous

$$\sum_{j=1}^{p} R_{ij} \leq 1, \quad \sum_{i=1}^{n} R_{ij} \leq 1 \quad \forall(i,j) \quad \text{(C.9a)}$$

$$\sum_{j=1}^{p} S_{jk} \leq 1, \quad \sum_{k=1}^{q} S_{jk} \leq 1 \quad \forall(j,k) \quad \text{(C.9b)}$$

$$\sum_{k=1}^{q} T_{ik} \leq 1, \quad \sum_{i=1}^{n} T_{ik} \leq 1 \quad \forall(i,k) \quad \text{(C.9c)}$$

$$R_{ij} + S_{jk} \leq T_{ik} + 1 \quad \forall(i,j,k) \quad \text{(C.9d)}$$

$$R_{ij} + T_{ik} \leq S_{jk} + 1 \quad \forall(i,j,k) \quad \text{(C.9e)}$$

$$S_{jk} + T_{ik} \leq R_{ij} + 1 \quad \forall(i,j,k) \quad \text{(C.9f)}$$

$$R_{ij} \in \{0,1\}, S_{jk} \in \{0,1\}, T_{ik} \in \{0,1\} \quad \forall(i,j,k) \quad \text{(C.9g)}$$

$$\text{(C.9h)}$$

où les contraintes (C.9d)-(C.9f) assurent la propriété (C.1).

Le problème devient plus simple dans le cas particulier où toutes les fonctions de masses de l'un des trois ensembles $\{m_{ij\cdot}\}$, $\{m_{\cdot jk}\}$ ou $\{m_{i\cdot k}\}$ sont vides. Par exemple, considérons que toutes les fonctions de masses $m_{i\cdot k}$ sont vides. C'est le cas où E, F et G sont les ensembles d'objets détectés par un seul capteur à des instants successifs, et nous calculons la similarité entres les objets détectés à deux temps consécutifs. Nous avons alors $w_{i\cdot k} = 0$ pour tout i et k et la fonction objective (C.8) devient une fonction de R et S seulement. Etant donné que les contraintes (C.9d)-(C.9f) peuvent être satisfaites pour une certaine relation T, en connaissant R et S, la fonction objective peut être maximisée par rapport à R et S séparément. Dans ce cas particulier, le problème tridimensionnel peut être résolu par la résolution de deux problèmes à deux dimensions comme démontré dans la section 5.4 du chapitre 5. Dans le cas général, le problème tridimensionnel est beaucoup plus complexe que le problème à deux dimensions, comme démontré dans la sous section suivante.

C.1.2. Analyse de complexité

Etant donné que le problème de la relation la plus plausible R^* in $\mathcal{R}$ est équivalent au problème d'affectation linéaire, trouver l'association la plus plausible $(R^*, S^*, T^*) \in \mathcal{U}$ est équivalent à un problème connu par *the axial 3-dimensional assignment problem* (3DAP) [Sch55], qui est $\mathcal{NP}$-difficile dans le cas général. Cependant, plusieurs classes particulières du problème sont polynômialement résoluble [Bc99]. 3DAP peut être considéré comme un cas particulier de notre extension tridimensionnelle avec $|E| = |F| = |G| = n$ et où R^*, S^*,T^* sont toutes des relations bijectives. Il est équivalent à trouver exactement n triplets $(e, f, g) \in E \times F \times G$ qui sont disjoints deux à deux. La différence majeure avec notre problème est que, lorsqu'on cherche l'association la plus plausible $(R^*, S^*, T^*) \in \mathcal{U}$, nous pouvons avoir $R^*_{ij} = 1$ alors qu'il n' ya pas un k tel que $S^*_{jk} = 1$ and $T^*_{ik} = 1$.

Même si nous parvenons à trouver une transformation du problème de l'association la plus plausible en 3 dimensions à 3DAP, nous n'avons pas de méthodes directes pour résoudre notre problème (comme 3DAP est $\mathcal{NP}$-Difficile).

Proposition 2. *Le problème d'association la plus plausible en 3-dimensions est $\mathcal{NP}$-difficile.*

Preuve: *Pour démontrer ce résultat, on utilise la réduction du* "pairwise consistent 3-dimensional matching problem" *[GJ79] noté par* $PC3DM$ *dans ce qui suit et qui est* $\mathcal{NP}$*-complet :*
INSTANCE : *Définir* $M \subseteq W \times X \times Y$*, où* W*,* X *et* Y *sont des ensembles disjoints ayant le même nombre d'éléments* r*.* M *est cohérent par paire, par exemple, pour tous les éléments* a*,* b*,* c*, chaque fois qu'il existe* w*,* x *et* y *tel que* $(a, b, y) \in M$*,* $(a, x, c) \in M$*, et* $(w, b, c) \in M$*, alors* $(a, b, c) \in M$*.*
QUESTION : *Est ce que* M *contient un sous ensemble* $M' \subseteq M$ *tel que* $|M'| = r$ *et il n'y a pas deux éléments de* M' *qui correspondent dans n'importe quelles coordonnées ?*

Etant donnée l'instance de $PC3DM$*, nous associons un cas d'un problème d'association à 3-dimensions de la manière suivante. Nous prenons* $E = W$*,* $F = X$ *et* $G = Y$ *et alors* $n = p = q = r$*. Pour tout* $(e_i, f_j, g_k) \in M$*, nous mettons* $w_{ij\cdot} = w_{i\cdot k} = w_{\cdot jk} = 1$*, alors que tous les autres poids sont mis à* 0*. Notons que cette réduction est polynomiale. Nous étudions la version de décision du problème d'association en 3 dimensions, dans lequel la question est de savoir s'il existe une association à 3 dimensions* $(R, S, T) \in \mathcal{U}$ *tel que la valeur de la fonction objective est de* $3r$*.*

Supposons qu'il existe un sous ensemble $M' \subseteq M$ *tel que* $|M'| = r$ *et il n'y a pas deux éléments de* M' *qui correspondent dans n'importe quelle cordonnée. Pour chaque* $(e_i, f_j, g_k) \in M'$*, nous prenons* $R_{ij} = S_{ik} = T_{jk} = 1$ *alors que les autres valeurs définissant les relations* R*,* S *et* T *sont mises à* 0*. Telles relations respectent les contraintes (C.1) et la valeur de la fonction objective est* $3r$ *puisque tous les* $3r$ *poids liés aux paires qui sont en relation prennent la valeur* 1*.*

Supposons maintenant qu'il existe une association à 3-dimensions $(R, S, T) \in \mathcal{U}$ *tel que la valeur de la fonction objective est* $3r$*. Par conséquent, chacun des* r *objets* $e_i \in E$ *est associé exactement à un seul objet* $f_j \in F$ *(*$e_i R f_j$*) avec* $w_{ij\cdot} = 1$ *et à un seul objet* g_k *in* G *(*$e_i T g_k$*) avec* $w_{i\cdot k} = 1$ *tel que* $f_j S g_k$ *est conservé et* $w_{\cdot jk} = 1$*. Pour chacune de ces* r *associations* e_i, f_j, g_k *définie par ces relations nous avons* $(e_i, f_j, g_k) \in M$*. En effet,* $w_{ij\cdot} = 1$ *implique que* $\exists y \in Y = G$ *tel que* $(e_i, f_j, y) \in M$*,* $w_{i\cdot j} = 1$ *implique que* $\exists x \in X = F$ *tel que* $(e_i, x, g_k) \in M$*,* $w_{\cdot jk} = 1$ *implique que* $\exists w \in W = E$ *tel que* $(w, e_i, g_k) \in M$*, et finalement ceci implique que* $(e_i, f_j, g_k) \in M$ *puisque* M *est cohérent deux à deux. Ainsi, il existe un sous ensemble* $M' \subseteq M$ *tel que* $|M'| = r$ *et il n'y a pas deux éléments qui correspondent en aucune cordonnée.* □

Ainsi il est peu probable de pouvoir résoudre le problème d'association à 3-dimensions en un temps polynomial. Pour des futurs travaux, nous pouvons adapter certaines

méthodes de la littérature de 3DAP pour résoudre notre problème. Nous pouvons alors utiliser les approches "branch and bound" (voir, par exemple, [BS91]) ou les algorithmes heuristiques (comme présenté dans [Pie67]) si le problème est trop compliqué pour être résolu d'un point de vue opérationnel.

Bibliographie

[air] Airplug : https ://www.hds.utc.fr/airplug/doku.php.

[AMO93] R. Ahuja, T. Magnanti, and J. Orlin. *Network flows : theory, algorithms and applications*. Prentice-hall, Boston, 1993.

[AS01] A. Ayoun and P. Smets. Data association in multi-target detection using the transferable belief model. *International Journal of Intelligent Systems*, 16 :1167–1182, 2001.

[ASKB12] M. Aeberhard, S. Schlichthärle, N. Kaempchen, and T. Bertramn. Track-to-track fusion with asynchronous sensors using information matrix fusion for surround environment perception. In *IEEE Intelligent Vehicles Symposium*, pages 1717–1726, Alcala de Henares, Spain, 3-7 June 2012.

[Bai02] T. Bailey. *Mobile Robot Localisation and Mapping in extensive outdoor environment*. PhD thesis, University of Sidney, 2002.

[Bc99] R. Burkard and E. Çela. *Handbook of Combinatorial Optimization - Supplement Volume A*. Kluwer Academic Publishers, 1999.

[BP99] S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.

[BS91] E. Balas and M. Saltzman. An algorithm for the three-index assignment problem. *Operations Research*, 39 :150–161, 1991.

[BS00] Y. Bar-Shalom. *Multitarget-Multisensor tracking : Applications and Advances*. Artech House, 2000.

[BSF88] Y. Bar-Shalom and T.E. Fortmann. *Tracking and data association*. Academic Press, Boston, 1988.

[BY02] M.P. Singh B. Yu. An evidential model of distributed reputation management. In *First international Joint Conference on Autonomous Agents and Multi-Agents Systems, ACM Press*, pages 294–301, Bologna, Italy, 2002.

[CAM02] L. Chen, P. O. Arambel, and R.K. Mehra. Estimation under unknown correlation : covariance intersection revisited. *IEEE Transactions on Automatic Control*, 47 :1879–1882, 2002.

[CDC08] V. Cherfaoui, T. Denoeux, and Z. L. Cherfi. Distributed data fusion : application to confidence management in vehicular networks. In *11th Int. Conf. on Information Fusion*, pages 846–853, Germany, 2008.

[CMHC13] J. Curn, D. Marinescu, N. O Hara, and V. Cahill. Data incest in cooperative localisation with the common past-invariant ensemble kalman filter. In *Proceedings of the 16th Int. Conf. on Information Fusion (FUSION '13)*, Istanbul, Turkey, 09-12 July 2013.

[com] Comosef :http ://www.celtic-initiative.org/projects/celtic-plus-projects/2011/comosef/comosef-default.asp.

[coo] Coopercom : http ://www.agence-nationale-recherche.fr/en/international-cooperation/recherches-exploratoires-et-emergentes/blanc-international/funded-project/.

[CV05] T. M. Chen and V. Venkataramanan. Dempster-shafer theory for intrusion detection in ad hoc networks. In *IEEE Internet Computing*, volume 9, pages 35–41, 2005.

[DCD12] B. Ducourthial, V. Cherfaoui, and T. Denoeux. Self-stabilizing distributed data fusion. *Stabilization, Safety, and Security of Distributed Systems , Lecture Notes in Computer Science*, 7596 :148–162, 2012.

[Dem67] A. P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *Annals of Mathematical Statistics*, 38 :325–339, 1967.

[Den95] T. Denoeux. A k-nearest neighbour classification rule based on dempster-shafer theory. *IEEE Transactions on Systems, Man and Cybernetics B*, 25(5) :804–813, 1995.

[Den00] T. Denoeux. A neural network classifier based on dempster-shafer theory. *IEEE Transactions on Systems, Man and Cybernetics A*, 30(2) :131–150, 2000.

[Den06] T. Denoeux. The cautious rule of combination for belief functions and some extensions. In *FUSION'2006*, Florence, Italy, July 2006.

[Den08] T. Denoeux. Conjunctive and disjunctive combination of belief functions induced by nondistinct bodies of evidence. *Artificial Intelligence*, 172 :234–264, 2008.

[DM04] T. Denoeux and M.-H. Masson. Evclus : Evidential clustering of proximity data. *IEEE transactions on Systems, Man and Cybernetics B*, 34(1) :95–109, 2004.

[DOO13] A. Dallil, M. Oussalah, and A. Ouldali. Sensor fusion and target tracking using evidential data associationl. *IEEE sensors journal*, 13(1) :285–293, 2013.

[Dou02] J.R Douceur. The sybil attack. In *the International Workshop on Peer to Peer Systems*, pages 251–260, Cambridge, MA, USA, 2002.

[dri] Drive-c2x : http ://www.drive-c2x.eu/project.

[Duc07] B. Ducourthial. r-semi-groups : A generic approach for designing stabilizing silent tasks. In *Self-Stabilizing Systems*, pages 281–295, 2007.

[EDG90] T. Einfalt, T. Denoeux, and G.Jacquet. A radar rainfall forecasting method designed for hydrological purposesl. *Journal of Hydrology*, 114 :229–244, 1990.

[FCD08] F. Fayad, V. Cherfaoui, and G. Derbhomez. Updating confidence indicators in a multi-sensor pedestrian tracking system. In *IEEE Intelligent Vehicles Symposium IV2008, Eindhoven*, 2008.

[GC99] D. Gruyer and V. Cherfaoui. Matching and decision for vehicle tracking in road situation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems IROS'99*, Kyongju, Korea, 17-21 October 1999.

[GD07] G. Guette and B. Ducourthial. On the sybil attack detection in vanet. In *International Workshop on Mobile Vehicular Networks (MoveNet 2007), co-located with IEEE MASS 2007*, Pisa, October 2007.

[GGS04] P. Golle, D. Greene, and J. Staddon. Detecting and correcting malicious data in vanets. In *1st ACM Workshop on Vehicular Ad hoc Networks (VANET)*, pages 29–37, New York, NY, USA, 2004.

[GJ79] M. Garey and D. Johnson. *Computers and intractability : a guide to the theory of NP-completeness*. W.H. Freeman, 1979.

[GRLD03] D. Gruyer, C. Royère, R. Labayrade, and D.Aubert. Credibilistic multi-sensor fusion for real time application, application to obstacle detection and tracking. In *IEEE Int. Conf. on Advanced Robotics, ICAR'2003*, pages 1463–1467, Coimbra, Portugal, June 3O-July 3 2003.

[HBCB12] A. Houenou, Ph. Bonnifait, V. Cherfaoui, and J.F. Boissou. A track-to-track association method for automotive perception systems. In *IEEE Intelligent Vehicles Symposium*, pages 704–710, Alcala de Henares, Spain, 3-7 June 2012.

[JKDW01] L. Jaulin, M. Kieffer, O. Didrit, and E Walter, editors. *Applied Interval Analysis*. Springer, 2001.

[JL08] K-H. Jo and J. Lee. Cooperative localization of multiple robots with constraint propagation technique. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3477–3482, Nice, France, September 2008.

[JU01] S. Julier and J. Uhlmann. *Handbook of Multidimensional Data Fusion (Eds. Hall D., Llians J.)*, chapter General decentralized data fusion with covariance intersection (CI). CRC Press, 2001.

[KCAC06] N. Karam, F. Chausse, R. Aufrère, and R. Chapuis. Localization of a group of communicating vehicles by state exchange. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2816–2821, Beijing, China, October 9-15 2006.

[kop] Ko-per- cooperative perception : http ://ko-fas.de/english/ko-per—cooperative-perception.html.

[Kuh55] H. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2 :83–97, 1955.

[LHL08] Martin E. Liggings, David L. Hall, and James Llinas, editors. *Handbook of multisensor Data Fusion Theory and practice, Second Edition (Electrical Engineering & Applied Signal Processing Series)*. CRC Press, 2008.

[LI04] J. Liu and V. Issarny. Enhanced reputation mechanism for mobile ad hoc networks. In *2nd International Conference on Trust Management*, pages 48–62, Oxford, UK, 2004.

[LN11] H. Li and F. Nashashibi. Multi-vehicle cooperative perception and augmented reality for driver assistance : A possibility to 'see' through front

vehicle. In *14th International IEEE Conference on Intelligent Transportation Systems*, pages 242–247, Washington, DC, USA, 05-07 October 2011.

[LN12] H. Li and F. Nashashibi. Cooperative multi-vehicle localization using split covariance intersection filter. In *IEEE Intelligent Vehicles Symposium*, pages 211–216, Madrid,Spain, 03 - 07 June 2012.

[MACC05] N. Meghrebi, S. Ambellouis, O. Colôt, and F. Cabestaing. Multimodal data association based on the use of belief functions for multiple target tracking. In *8th International Conference on Information Fusion (FUSION)*, pages 900–906, Philadelphia, PA (USA), 2005.

[Mit07] H.B. Mitchell. *Multisensor Data Fusion : An introduction*. Springer, 2007.

[MKC03] S. McLaughlin, V. Krishnamurthy, and S.l Challa. Managing data incest in a distributed sensor network. In *Proceedings IEEE International Conference on Accoustics, speech and Signal Processing 5*, 6-10 April 2003.

[MKE04] S. Mclaughlin, V. Krishnamurthy, and R. J. Evans. Bayesian network model for data incest in a distributed sensor network. In *the 7 th International Conference on Information Fusion*, volume 1, Stockholm, Sweden, 2004.

[MLJ11] D. Mercier, E Lefèvre, and D. Jolly. Object association with belief functions, an application with vehicles. *Information Sciences*, 181(24) :5485–5500, December 2011.

[MPS05] A. Martinelli, F. Pont, and R. Siegwart. Multi-robot localization using relative observations. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 2808–2813, Barcelona, Spain, April 2005.

[OA92] J. Orlin and R. Ahuja. New scalin algorithms for the assignment and minimum cucle means problems. *Mathematical programming*, 54 :41–56, 1992.

[pac] Pacpus : http ://www2.hds.utc.fr/pacpus.

[Pie67] W. Pierskalla. The tri-substitution method for the three-multidimensional assignment problem. *Canadian ORS Journal*, 5 :71–81, 1967.

[PSL06] C. Piro, C. Shields, and B.N Levine. Detecting the sybil attack in mobile ad hoc networks. In *IEEE/ACM Intl Conf on Security and privacy in Communication Networks (SecureComm)*, pages 1–11, August 2006.

[RB02] S. Roumeliotis and G. A. Bekey. Distributed multirobot localization. *IEEE Transactions on Robotics and Automation*, 18(5) :781–795, 2002.

[RC97] M. Rombaut and V. Cherfaoui. Decision making in data fusion using dempster-shafer's theory. In *3th IFAC Symposium on Intelligent Components and Instrumentation for Control Applications*, Annecy, France, 9-11 june 1997.

[Rom98] M. Rombaut. Decision in multi-obstacle matching process using the theory of belief. In *Advances in Vehicle Control and safety, AVCS98*, pages 63–68, Amiens, France, 1-3 july 1998.

[RPGH08] M. Raya, P. Papadimitratos, V. D. Gligor, and J-P. Hubaux. On data-centric trust establishment in ephemeral ad hoc networks. In *the 28th IEEE conference on Computer Communications (INFOCOM)*, pages 1238–1246, Phoenix, AZ., USA, April 2008.

[RS07] B. Ristic and P. Smets. Global cost of assignment in the tbm framework for association of uncertain id reports. *Aeorspace Science and Technology*, 11(4) :303–309, 2007.

[saf] The safespot project : http ://www.safespot-eu.org/.

[SAR+12] R. Schubert, C. Adam, E. Richter, S. Bauer, H. Lietz, and G. Wanielik. Generalized probabilistic data association for vehicle tracking under clutter. In *Intelligent Vehicles Symposium*, pages 962–968, Alcala de Henares, Spain, 3-7 June 2012.

[Sch55] E. Schell. Distribution of a product by several properties, directorate of management analysis. In *proceedings of the Second Symposium in Linear Programming*, pages 615–642, DCS/Comptroller H.Q. U.S.A.F., Washington, DC., 1955.

[Sch00] J. Schubert. Managing inconsistent intelligence. In *3th Int. Conf. on Information Fusion*, pages 4–10, Paris, France, 2000.

[Sch08] J. Schubert. Clustering decomposed belief functions using generalized weights of conflict. *International Journal of Approximate Reasonning*, 48(2) :466–480, 2008.

[Sha76] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[SK94] P. Smets and R. Kennes. The transferable belief model. *Artificial Intelligence*, 66 :191–234, 1994.

[Sme93a] P. Smets. Belief functions : The disjunctive rule of combination and the generalized bayesian theorem. *International Journal of Approximate Reasoning*, 9 :1–35, 1993.

[Sme93b] P. Smets. Belief functions : the disjunctive rule of combination and the generalized Bayesian theorem. *International Journal of Approximate Reasoning*, 9 :1–35, 1993.

[Sme00] P. Smets. Data fusion in the transferable belief model. In *3rd International Conference on Information Fusion*, 2000.

[Smi06] K. Smith. Reversible-jump markov chain monte carlo multi-object tracking tutorial. Communication IDIAP-COM-06-07, IDIAP Research Institute, 2006.

[TB04] G. Theodorakopoulos and J. S. Baras. Trust evaluation in ad-hoc networks. In *ACM Workshop Wireless Security*, pages 1–10, Philadelphia, PA, USA, 2004.

[TH05] K. Tischler and B. Hummel. Enhanced environmental perception by inter-vehicle data exchange. In *IEEE Intelligent Vehicles Symposium*, pages 313–318, Las Vegas,USA, 06 - 08 June 2005.

[uru] Urus project : http ://www.urus.upc.es/nuevoperception.html.

[WS09] J. Wang and H-J. Sun. A new evidential trust model for open communities. *Computer Standards & Interfaces*, 31 :994–1001, 2009.

[XBG06] B. Xiao, B.Yu, and C. Gao. Detection and localization of sybil nodes in vanets. In *the Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks*, pages 1–8, Los Angeles, CA,USA, 2006.

[YCWO08] G. Yan, G. Choudhary, M. Weigle, and S. Olariu. Providing vanet security through active position detection. *Computer Communications : Special Issue on Mobility Protocols for ITS/ VANET*, 31(12) :2883–2897, 2008.

[YXX13] B. Yu, C-Z. Xu, and B. Xiao. Detecting sybil attacks in vanets. *Journal of Parallel and Distributed Computing*, 73 :746–756, 2013.

[ZM00] G. Zacharia and P. Maes. Trust management through reputation mechanisms. *Applied Artificial Intelligence*, 14 :881–907, 2000.