



# UNIVERSITÉ FRANÇOIS RABELAIS DE TOURS

École Doctorale MIPTIS  
LABORATOIRE : LI EA 6300

**THÈSE** présentée par :  
**Djibrilla Amadou Kountché**

pour obtenir le grade de : Docteur de l'Université François Rabelais - Tours  
Discipline : Informatique

**Localisation dans les bâtiments des personnes handicapées et  
classification automatique de données par fourmis artificielles**

**THÈSE DIRIGÉE PAR :**

MONMARCHÉ Nicolas

Maître de Conférences, Université François Rabelais - Tours

SLIMANE Mohamed

Professeur, Université François Rabelais - Tours

**RAPPORTEURS :**

FONLUPT Cyril

Professeur, Université du Littoral - Côte d'Opale

SIARRY Patrick

Professeur, Université Paris-Est Créteil Val de Marne - Université Paris 12

**JURY :**

FONLUPT Cyril

Professeur, Université du Littoral - Côte d'Opale

JAIME LÓPEZ Krahe

Professeur, Université Paris 8

MONMARCHÉ Nicolas

Maître de Conférence, Université François Rabelais - Tours

SIARRY Patrick

Professeur, Université Paris-Est Créteil Val de Marne - Université Paris 12

SLIMANE Mohamed

Professeur, Université François Rabelais - Tours

TREUILLET Sylvie

Maître de conférences, Université d'Orléans



# Remerciements



# Résumé

Le concept du « smart » envahit de plus en plus notre vie quotidienne. L'exemple type est sans doute le smartphone. Celui-ci est devenu au fil des ans un appareil incontournable. Bientôt, c'est la ville, la voiture, la maison qui seront « intelligentes ».

L'intelligence se manifeste par une capacité d'interaction et de prise de décision entre l'environnement et l'utilisateur. Ceci nécessite des informations sur les changements d'états survenus des deux côtés. Les réseaux de capteurs permettent de collecter ces données, de leur appliquer des pré-traitements et de les transmettre aux applications. Ces réseaux de par certaines de leurs caractéristiques se rapprochent de l'intelligence collective, dans le sens, où des entités de faibles capacités se coordonnent automatiquement, sans intervention humaine, de façon décentralisée et distribuée pour accomplir des tâches complexes. Ces méthodes bio-inspirées ont servi à la résolution de plusieurs problèmes, surtout l'optimisation, ce qui nous a encouragé à étudier la possibilité de les utiliser pour les problèmes liés à l'*Ambient Assisted Living* ou AAL et à la classification automatique de données.

L'AAL est un sous-domaine des services dits basés sur le contexte, et a pour objectifs de faciliter la vie des personnes âgées et handicapées dans leurs défis quotidiens. Pour ce faire, il détermine le contexte et, sur cette base, propose divers services. Deux éléments du contexte nous ont intéressés : le handicap et la position.

Bien que la détermination de la position (localisation, positionnement) se fasse à l'extérieur des bâtiments avec des précisions très satisfaisantes, elle rencontre plusieurs difficultés à l'intérieur des bâtiments, liées à la propagation des ondes électromagnétiques dans les milieux difficiles, aux coûts des systèmes, à l'interopérabilité, etc.

Nos travaux se sont intéressés au positionnement des personnes handicapées à l'intérieur de bâtiments en utilisant un réseau de capteurs afin de déterminer les caractéristiques de l'onde électromagnétique (puissance, temps, angle) pour estimer la position par méthodes géométriques (triangulation, latération), méthodes de fingerprinting (k plus proches voisins), par des filtres baysiens (filtre de Kalman). L'application est d'offrir des services types AAL tel que la navigation. Nous avons élargi la notion de réseau de capteurs pour prendre en compte tout appareil capable d'émettre et de recevoir une onde électromagnétique et se trouvant dans l'environnement. Nous avons aussi appliqué l'algorithme API sur la classification automatique de données. Enfin, nous avons proposé une architecture à middleware pour la localisation *indoor*.

**Mots clés :** Réseaux de capteurs, Intelligence Collective, Clustering, Positionnement *indoor*, Ambient Assisted Living, Personnes âgées et handicapées.



# Abstract

The concept of « smart » invades more and more our daily life. A typical example is the smartphone, which becomes by years an essential device. Soon, it's the city, the car and the home which will become « smart ». The intelligence is manifested by the ability for the environment to interact and to take decisions in its relationships with users and other environments. This needs information on state changes occurred on both sides. Sensor networks allow to collect these data, to apply on them some pre-processings and to transmit them. Sensor network, towards some of their characteristics are closed to Swarm Intelligence in the sense that small entities with reduced capabilities can cooperate automatically, in unattended, decentralised and distributed manner in order to accomplish complex tasks. These bio-inspired methods have served as basis for the resolution of many problems, mostly optimization and this inspired us to apply them on problems met in Ambient Assisted Living and on the data clustering problem.

AAL is a sub-field of context-aware services, and its goals are to facilitate the everyday life of elderly and disable people. These systems determine the context and then propose different kind of services. We have used two important elements of the context : the position and the disability. Although positioning has very good precision outdoor, it faces many challenges in indoor environments due to the electromagnetic wave propagation in harsh conditions, the cost of systems, interoperability, etc.

Our works have been involved in positioning disabled people in indoor environment by using wireless sensor network for determining the characteristics of the electromagnetic wave (signal strength, time, angle) for estimating the position by geometric methods (triangulation, lateration), fingerprinting methods (k-nearest neighbours), bayesian filters (Kalman filter). The application is to offer AAL services like navigation. Therefore we extend the definition of sensor node to take into account any device, in the environment, capable of emitting and receiving a signal. Also, we have studied the possibility of using *Pachycondylla Apicalis* for data clustering and for indoor localization by casting this last problem as data clustering problem. Finally we have proposed a system based on a middleware architecture.

**Keywords :** Wireless Sensor Networks, Swarm intelligence, Localization, Positioning, Data Clustering, Ambient Assisted Living, Context-Aware services, Location-based services, Disabled and elderly people.

## ABSTRACT

---

# Table des matières

<b>Introduction</b>	<b>17</b>
<b>I État de l'art de la localisation <i>indoor</i> par réseaux de capteurs</b>	<b>25</b>
<b>1 Introduction aux réseaux de capteurs</b>	<b>27</b>
1.1 Introduction . . . . .	27
1.2 Présentations . . . . .	27
1.3 Caractéristiques . . . . .	29
1.4 Architecture d'un capteur . . . . .	30
1.4.1 L'unité de traitement . . . . .	32
1.4.2 Unité de mesure . . . . .	32
1.4.3 L'unité de communication . . . . .	34
1.4.4 Extension de la définition d'un capteur . . . . .	34
1.5 La communication dans les réseaux de capteurs . . . . .	35
1.5.1 La norme IEEE 802.11 (ISO/IEC 8802-11) . . . . .	35
1.5.2 La norme IEEE 802.15 (Bluetooth) . . . . .	39
1.5.3 La norme IEEE 802.15.4 . . . . .	41
1.5.4 Zigbee . . . . .	43
1.5.5 RFID/NFC . . . . .	45
1.6 Modélisation des réseaux de capteurs . . . . .	47
1.6.1 Modèle du Graphe Unitaire ( <i>Unit Disk Graph</i> ) . . . . .	48
1.6.2 Propriétés . . . . .	48
1.6.3 Bilan de la liaison . . . . .	50
1.6.4 Modélisation déterministe réaliste . . . . .	52
1.6.5 Affaiblissement moyen . . . . .	52
1.6.6 Modélisation stochastique . . . . .	52
1.6.7 Les évanouissements non sélectifs . . . . .	55
1.7 Simulation des réseaux de capteurs . . . . .	55

1.8	Conclusion . . . . .	56
<b>2</b>	<b>La localisation par réseaux de capteurs</b>	<b>59</b>
2.1	Introduction . . . . .	59
2.2	Définitions . . . . .	60
2.3	Les techniques de bases de mesures . . . . .	62
2.3.1	La Mesure de la Puissance du Signal . . . . .	63
2.3.2	Le temps d'arrivée . . . . .	64
2.3.3	L'angle d'arrivée . . . . .	65
2.3.4	La Différence entre Temps d'Arrivée . . . . .	66
2.3.5	Le Profilage de la Puissance du Signal . . . . .	66
2.4	Les méthodes de positionnement . . . . .	67
2.4.1	Les méthodes géométriques . . . . .	68
2.4.2	Les techniques de proximité . . . . .	74
2.4.3	Les méthodes de fingerprinting . . . . .	75
2.4.4	Les méthodes bayésiennes . . . . .	77
2.5	Conclusion . . . . .	80
<b>II</b>	<b>Mise en oeuvre de la localisation de personnes handicapées</b>	<b>81</b>
<b>3</b>	<b>Étude des Systèmes Ambient Assisted Living</b>	<b>83</b>
3.1	Introduction . . . . .	83
3.2	La localisation des personnes à mobilité réduite . . . . .	83
3.3	État de l'art de systèmes de AAL . . . . .	84
3.4	Principes de conception d'un système de localisation . . . . .	87
3.4.1	Exactitude ( <i>accuracy</i> ) / Précision ( <i>precision</i> ) . . . . .	88
3.4.2	La couverture ( <i>coverage</i> ) . . . . .	89
3.4.3	L'intégrité ( <i>integrity</i> ) . . . . .	90
3.4.4	La disponibilité ( <i>availaility</i> ) . . . . .	90
3.4.5	La continuité . . . . .	90
3.4.6	Le taux de mise à jour ( <i>Update Rate</i> ) . . . . .	90
3.4.7	Le temps de latence du système . . . . .	91
3.4.8	Les données de sortie ( <i>Output Data</i> ) . . . . .	91
3.4.9	Les interfaces homme-machine ( <i>Interface</i> ) . . . . .	91
3.4.10	Sécurité et Préservation de la vie privée . . . . .	91
3.4.11	Les coûts . . . . .	92
3.5	Processus de définition des besoins utilisateurs . . . . .	92

3.6	Conclusion . . . . .	93
<b>4</b>	<b>AIPS : Vers Un Autre Système de Géolocalisation Interne et d'Assistance à la Mobilité</b>	<b>95</b>
4.1	Introduction . . . . .	95
4.2	Première proposition . . . . .	95
4.2.1	L'application Superviseur . . . . .	96
4.2.2	Le Réseau de Capteurs . . . . .	99
4.2.3	Le Smartphone . . . . .	103
4.2.4	Expérimentations . . . . .	103
4.3	Seconde proposition . . . . .	111
4.3.1	La Couche Réseau . . . . .	111
4.3.2	Le Middleware . . . . .	112
4.3.3	Localisation . . . . .	113
4.4	Conclusion . . . . .	114
<b>III</b>	<b>Étude et Application de la classification par algorithmes de four-</b>	<b>115</b>
	<b>mis</b>	
<b>5</b>	<b>La classification automatique de données</b>	<b>117</b>
5.1	Introduction . . . . .	117
5.2	Le processus du clustering . . . . .	118
5.2.1	Le prétraitement des données . . . . .	118
5.2.2	Définitions . . . . .	119
5.3	Le problème étudié . . . . .	123
5.4	Les méthodes de partitionnement . . . . .	124
5.4.1	Évaluation et Comparaison de classes . . . . .	125
5.4.2	Évaluation d'une partition . . . . .	127
5.4.3	Comparaison de partitions . . . . .	128
5.4.4	k-means : un exemple d'algorithme de partitionnement . . . . .	131
5.4.5	La détermination du nombre de classes $k$ . . . . .	132
5.5	Les méthodes hiérarchiques . . . . .	132
5.5.1	Évaluation d'une hiérarchie . . . . .	136
5.6	Les méthodes probabilistes . . . . .	136
5.7	Les méthodes basées sur la densité . . . . .	138
5.7.1	DBSCAN . . . . .	138
5.7.2	DENCLUE . . . . .	140
5.8	La classification par fourmis artificielles . . . . .	141

## TABLE DES MATIÈRES

---

5.8.1	Les origines . . . . .	141
5.8.2	Le modèle de Deneubourg . . . . .	143
5.8.3	Le modèle de Lumer et Faieta . . . . .	144
5.9	Conclusion . . . . .	147
<b>6</b>	<b>API : Les méthodes de chasses des fourmis <i>Pachycondyla apicalis</i> appliquées au clustering</b>	<b>149</b>
6.1	Introduction . . . . .	149
6.2	API pour la classification de données . . . . .	149
6.2.1	Principes de API . . . . .	149
6.2.2	Notations et conditions initiales . . . . .	150
6.2.3	Description du comportement des agents . . . . .	151
6.3	Expérimentations . . . . .	153
6.3.1	Description des données . . . . .	155
6.3.2	Les fonctions d'évaluation . . . . .	155
6.3.3	Les résultats . . . . .	158
6.4	Vers une version non supervisée de API . . . . .	166
6.5	Les fourmis <i>P. Apicalis</i> pour la localisation interne . . . . .	167
6.6	Conclusion . . . . .	167
	<b>Conclusion</b>	<b>169</b>
	<b>Annexes</b>	<b>177</b>
	<b>A Description des données aléatoires</b>	<b>177</b>

# Liste des tableaux

1.1	Caractéristiques des MCU MSP430F123 et ATMEGA128L . . . . .	32
1.2	Exemple de grandeurs physiques . . . . .	33
1.3	Couches des protocoles IEEE 802.11 . . . . .	36
1.4	Normes physiques IEEE 802.11 . . . . .	37
1.5	Couches des protocoles IEEE 802.15 . . . . .	39
1.6	Classes des <i>transceivers</i> Bluetooth . . . . .	40
1.7	couche des protocoles IEEE 802.15.4 . . . . .	42
1.8	Caractéristiques de la transmission RFID . . . . .	45
1.9	Les types de tags NFC . . . . .	46
1.10	Mode de communication NFC-A, NFC-B, NFC-F . . . . .	46
1.11	Valeurs pour la variance $\sigma^2$ de $X_\sigma$ . . . . .	54
3.1	Les divers types de déploiement réguliers avec les couvertures pour chaque nœud. . . . .	89
4.1	Table d'association simple de RSSI et de Distance pour un point de référence.	102
5.1	Les matrices d'objets et de dissimilarité . . . . .	119
6.1	Les bases de données artificielles. . . . .	156
6.2	Un exemple de matrice de confusion . . . . .	156
6.3	Résultats obtenus avec API et les bases artificielles pour $\alpha = 0.1$ . . . . .	158
6.4	Résultats obtenus avec API et les bases artificielles pour $\alpha = 0.5$ . . . . .	158
6.5	Résultats obtenus avec API et les bases artificielles pour $\alpha = 1$ . . . . .	159
6.6	Résultats obtenus avec API et les bases artificielles pour $\alpha = 10$ . . . . .	159
6.7	Résultats obtenus avec API et les bases artificielles pour $\alpha = 100$ . . . . .	160
6.8	Résultats obtenus avec API et art5 pour $\alpha = 05$ . . . . .	160
6.9	Résultats obtenus avec API et art5 pour $\alpha = 05$ . . . . .	162
6.10	Résultats obtenus avec API et art6 pour $k_p$ . . . . .	162
6.11	Résultats obtenus avec API et art6 pour $k_p$ . . . . .	162

## LISTE DES TABLEAUX

---

6.12	Comparaison API K-means pour Iris et Wine . . . . .	165
6.13	Résultats obtenus avec API et K-means pour Iris et Wine pour $k = 4$ . . . . .	165
6.14	Résultats obtenus avec API et K-means, ACA pour Iris. . . . .	166
6.15	Résultats obtenus avec API et K-means, ACA pour Iris . . . . .	166
A.1	Les bases de données artificielles. . . . .	177

# Table des figures

1.1	Réseau de capteurs sans fil corporel . . . . .	28
1.2	Architecture d'un capteur . . . . .	31
1.3	Capteurs de Gaz et d'événements . . . . .	33
1.4	Exemple de modules de transmission . . . . .	34
1.5	Modes de fonctionnement d'un réseau IEEE 802.11 . . . . .	37
1.6	Architecture d'un réseau Bluetooth . . . . .	41
1.7	Spécification de la pile de protocoles Zigbee . . . . .	44
2.1	Positionnement <i>outdoor</i> de vaches . . . . .	61
2.2	Positionnement <i>indoor</i> de personnes . . . . .	61
2.3	Positionnement <i>indoor</i> d'un humanoïde . . . . .	61
2.4	Le processus du Positionnement . . . . .	63
2.5	Mesure du temps d'arrivée . . . . .	65
2.6	Techniques de mesure AOA et TDOA . . . . .	66
2.7	Cas de l'intersection de 3 cercles . . . . .	69
2.8	La méthode Min-Max . . . . .	74
3.1	La plateforme Aladdin . . . . .	86
3.2	L'architecture de SM4All . . . . .	87
3.3	Définition des besoins des utilisateurs . . . . .	92
4.1	Illustration du prototype . . . . .	96
4.2	Architecture du prototype . . . . .	96
4.3	Versions de XBEE . . . . .	100
4.4	Coordinateur Waspnote . . . . .	100
4.5	Routeur Waspnote . . . . .	101
4.6	Diagramme des expérimentations . . . . .	104
4.7	Le plan de l'environnement d'expérimentation . . . . .	108
4.8	Le plan en 3D . . . . .	109

## TABLE DES FIGURES

---

4.9	La couche réseau . . . . .	112
4.10	Le middleware . . . . .	112
5.1	AGNES et DIANA . . . . .	133
5.2	Exemple de hiérarchie . . . . .	134
5.3	Une hiérarchie indicée . . . . .	134
5.4	illustration DBSCAN . . . . .	139
6.1	Grille utilisée par API . . . . .	151
6.2	Tableau des résultats de API . . . . .	159
6.3	Tableau des résultats de API . . . . .	160
6.4	Illustration du clustering de la base art2 . . . . .	161
6.5	Diagramme des expérimentations pour API . . . . .	173

# Introduction

Les nouvelles technologies, particulièrement les récentes avancées dans les technologies de transmission sans fil ont changé notre vie de tous les jours. Les travaux que nous avons entrepris rentrent dans le cadre de la promotion de ces changements pour les personnes handicapées.

*« Une personne handicapée est une personne qui, du fait d'une déficience physique ou mentale, ne peut exercer certaines activités ou participer à la vie sociale. On peut distinguer six grands types de handicaps : moteur, auditif, visuel, psychique, intellectuel et lié à une maladie. Dans ses enquêtes sur le handicap, l'Insee distingue des limitations motrices, sensorielles et cognitives. L'institut distingue le handicap « ressenti » (se déclarer handicapé quel que soit le handicap), le handicap « identifié » (en fonction de critères définis par l'enquête) et le handicap « reconnu » (par une administration). La notion de handicap est relative et évolue dans le temps : elle dépend des normes sociales en vigueur. L'ampleur de la population handicapée varie selon le niveau de déficience considéré. Le handicap ouvrant des droits (dits « droits à compensations »), il fait aussi l'objet d'une reconnaissance administrative. Concrètement, une personne handicapée doit faire une demande auprès de l'une des Maisons départementales des personnes handicapées (structures créées par la loi de février 2005, animées par le Conseil général) qui dresse un bilan et établit un « plan personnalisé de compensation » » [DILA, 2013].*

En France, c'est la loi de février 2005, dite « loi pour l'égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées » qui offre un cadre législatif contraignant pour l'établissement des dispositifs, facilitant la vie de ces personnes.

Elle vise cinq objectifs :

1. *« la création d'un droit à compensation qui permet la prise en charge par la collectivité des dépenses liées au handicap (aide humaine et technique). Le texte précise que le versement de cette prestation, dite " universelle ", soumis à des conditions d'âge et de revenu, sera progressivement élargi, dans un délai de 3 à 5 ans, à toute personne handicapée quels que soient son âge et son salaire ;*
2. *l'intégration scolaire, en posant le principe de la scolarisation dans l'établissement le plus proche pour les enfants handicapés et la mise en place des aménagements nécessaires pour permettre aux étudiants handicapés une scolarité continue ainsi que le droit à une évaluation régulière de leurs compétences et de leurs besoins ;*
3. *l'insertion professionnelle : la loi prévoit un système d'incitations et de sanctions alourdies pour faire respecter l'obligation légale de l'emploi d'un quota de 6% de personnes handicapées dans le secteur public et dans les entreprises privées de plus*

*de vingt personnes ;*

4. *le renforcement de l'accessibilité, qui concerne l'accès des personnes handicapées aux espaces publics, aux systèmes de transport et au cadre bâti neuf. Des dispositifs d'incitation et de sanction sont également prévus ;*
5. *la simplification administrative, avec la création de maisons départementales des personnes handicapées où les différentes commissions préexistantes seront fusionnées en un " guichet unique ", la commission des droits et de l'autonomie des personnes handicapées. » [Observatoire Société, 2013]*

Au niveau Européen, le handicap et le vieillissement de la population deviennent des enjeux majeurs, du fait, entre autre, des coûts liés aux soins, et aussi le besoin de faire contribuer ces personnes à la vie socio-économique. Cependant, chaque pays a ses spécificités dans la prise en charge des personnes âgées et handicapées. L'étude de Hélène Gallo a montré qu'une homogénéisation est en cours, depuis des années, sous l'égide de l'Union Européenne, entre les diverses politiques.

Au niveau mondial, la population vieillit plus vite (1,9%) qu'elle n'augmente (1,02%). Ce pourcentage devrait plus que tripler pour les 50 années à venir [ERCIM, 2011]. Et, pour certains pays comme la Grande Bretagne, les États-Unis, l'Italie, l'Allemagne un plus fort taux d'accroissement de la population âgée est attendu du fait, entre autre des « babyboomers » et du faible taux de natalité.

Ainsi, le besoin se fait partout sentir de faire plus travailler les personnes âgées et handicapées, de ne pas les laisser s'isoler du fait des difficultés, de tirer parti de leur expérience au travers des interactions sociales, etc. Les nouvelles technologies peuvent y contribuer fortement. Plusieurs approches existent allant, du simple apprentissage de l'usage de ces technologies, à la mise en place de systèmes intelligents capables de suivre et de proposer des services adaptés. Cette catégorie de services est connue sous le nom de « services basés sur le contexte ». De plus en plus d'approches ne considèrent pas l'approche technologique comme la panacée, mais comme un outil complémentaire à d'autres approches.

Le contexte est un concept très large comportant toutes les caractéristiques de l'individu, et d'autres informations spatiales, temporelles, sociales, etc. permettant de déterminer une situation particulière et d'adapter l'environnement à l'utilisateur. Ainsi par exemple, une maison peut, après avoir déterminé qu'une personne a un malaise, appeler les services de soins et les guider jusqu'à cette personne.

Deux éléments du contexte nous ont intéressé : la position et le handicap. Nous avons commencé par introduire ce que c'est que le handicap et un certain nombre d'objectifs fixés pour les systèmes destinés à ces personnes. Quant à la position, c'est un des éléments fondamentaux d'un contexte. Elle permet de bâtir des services les plus variés et des plus importants. Cependant, sa détermination n'est pas encore totalement satisfaisante au vue des attentes. Si c'est le cas, c'est avec un coût élevé lié à l'acquisition d'une technologie.

Le positionnement, ou localisation, consiste à déterminer la *position* d'une *entité* connaissant les *estimations* de ses relations avec certains points de référence ou d'autres entités.

La relation entre nœuds peut être estimée suivant plusieurs méthodes. Par exemple, le changement de paramètres dans une zone donnée dont la position est connue. Le changement de température, de luminosité dans une chambre ou la mise en marche de la télévision peuvent permettre d'inférer une position et un contexte donné. Trois grandes catégories

peuvent être définies pour l'obtention de données pour le positionnement :

1. inertielle : données obtenues par des accéléromètres, gyroscopes et par la technique de mesure dite *dead reckoning* ;
2. ondes mécaniques : c'est le cas du signal sonore et ultrason ;
3. ondes électromagnétiques. C'est la catégorie la plus utilisée par exemple : les radars, les réseaux cellulaires, les téléphones *Digital Enhanced Cordless Telephone* (DECT), l'ultra large bande (ULB), les systèmes GNSS et les systèmes pseudo-satellites. Dans la suite de ce document, nous n'allons nous focaliser que sur une partie du spectre électromagnétique. Soit un signal :

$$s(t) = a \sin(2\pi ft + \phi) \quad (1)$$

Où  $a$  est l'amplitude du signal,  $f$  sa fréquence et  $\phi$  sa phase. Ainsi nous allons présenter des techniques de mesures permettant d'estimer les valeurs de ces caractéristiques en se basant sur des modèles de propagation du signal.

Le positionnement peut se dérouler dans les environnements suivants avec des contraintes différentes :

- *outdoor* : principalement en espace ouvert ;
- *indoor* : les espaces clos : maisons, hangars, centres commerciaux ;
- hybride : système opérant dans les deux milieux.

Le positionnement *outdoor* dispose d'une technologie *de facto* qui est basée sur les systèmes de navigation satellites (*Global Navigation Satellite Systems* ou GNSS) dont le GPS est le plus utilisé. D'autres systèmes existent, tels que l'européen Galiléo et le russe GLONASS. Cependant, ces systèmes GNSS ne fonctionnent pas dans les environnements difficiles c'est-à-dire ayant une forte densité d'obstacles et s'ils y fonctionnent c'est avec des performances dégradées.

Le positionnement *indoor* a pour objectifs de fournir une position là où les systèmes *outdoor* ne fonctionnent pas avec pour objectifs de baisser les coûts des systèmes et surtout de prendre en compte certaines caractéristiques des environnements *indoor*. Ces caractéristiques sont [Mautz, 2012] :

- des signaux multi-trajets provenant de la réflexion du signal sur les obstacles ;
- les conditions de non propagation en lignes directes (*Non Line of Sight*) ;
- forte atténuation et une forte dispersion dues à la présence d'obstacles ;
- rapides changements temporels dus à la présence de personnes et à l'ouverture des portes ;
- forte demande en précision et en exactitude pour certaines applications.

Cependant, d'autres caractéristiques des environnements *indoor* contribuent à faciliter le positionnement et ce de plusieurs manières [Mautz, 2012] :

- présence de zones de petites couvertures ;
- faible influence du climat par exemple faible gradient de température<sup>1</sup> et lente circulation de l'air ;

---

1. Le gradient de température,  $\vec{\nabla}T$ , est une grandeur indiquant la variation de la température en un point. Une faible variation de la température est une bonne condition pour l'usage des ondes mécaniques qui sont perturbées par l'augmentation de la température.

- contraintes géométriques fixes ;
- disponibilité d'autres infrastructures comme par exemple les réseaux électriques, les réseaux locaux et des murs adaptés pour le montage des nœuds de référence ;
- faibles dynamiques dues à une marche lente et une faible vitesse.

Le positionnement hybride se déroule dans les deux milieux. Son fonctionnement peut être l'usage combiné de deux systèmes de positionnement distincts adaptés pour chaque milieu ou un même système capable de fonctionner *indoor/outdoor*.

Dans les trois cas, une fois que les informations de positionnement sont obtenues, des algorithmes sont utilisés pour déterminer la position. Ces méthodes peuvent être regroupées en plusieurs catégories que nous allons détailler dans le Chapitre 2.

Le problème de la localisation à l'intérieur comme à l'extérieur de bâtiment peut aussi être modélisé en problème d'optimisation ou de classification supervisée ou non et de ce fait plusieurs algorithmes ont été utilisés depuis Doherty [Doherty *et al.*, 2001] jusqu'aux méthodes bio-inspirées : PSO [Kulkarni et Kumar, 2011] par exemple.

La classification de données s'intéresse à déterminer des classes dans des bases de données selon les approches suivantes : i) la classification supervisée ; ii) la classification non supervisée ou *clustering* et iii) la classification semi-supervisée.

La classification supervisée utilise les classes déjà connues pour déterminer les classes dans lesquelles les nouveaux objets seront placés. La problématique de la classification non supervisée est la suivante : étant donné un certain nombre d'objets, sur lesquels peu ou prou d'informations sont disponibles, est-il possible d'identifier des classes dans ces données et s'il y en a le nombre exact ou un nombre de classes approprié pour les besoins de l'utilisateur. La classification semi-supervisée est une approche intermédiaire entre les deux précédentes qui consiste à utiliser des informations a priori pour déterminer les classes. Un exemple est la méthode des k-moyennes, pour laquelle le nombre de classe à déterminer est fixé à l'avance. Ces trois approches peuvent être hybridées. Par exemple, le résultat d'une méthode semi-supervisée peut servir d'entrée à une méthode supervisée.

Nous nous sommes intéressés à la classification non-supervisée ou *clustering*. Selon les hypothèses sur les classes à trouver, nous distinguons les méthodes :

- hiérarchiques : elles construisent une hiérarchie sur les données. Les classes sont déterminées en « coupant » à un moment donné le dendrogramme construit par l'algorithme ;
- de partitionnement : l'hypothèse est qu'il existe un nombre  $k$  de classes. Les données sont itérativement ré-allouées à des classes jusqu'à ce qu'un critère d'arrêt soit atteint. Une variante est de relâcher la contrainte d'appartenance d'une donnée à une seule classe, de sorte qu'une donnée appartienne à plusieurs classes en même temps mais avec une probabilité. Au final, la donnée sera allouée à la classe de plus forte probabilité ;
- probabilistes : l'hypothèse est que les données ont été générées à partir d'une ou plusieurs mixtures de loi de probabilité. La classification consiste à estimer les paramètres de ces lois à partir de la base de données. Cependant, le nombre de classes à trouver peut être soit fixé à l'avance ou estimé durant une phase préliminaire ;
- de densité : l'hypothèse est que les classes sont des régions denses séparées par des régions moins denses. Une recherche est effectuée dans le voisinage des données pour

déterminer ces régions.

Parmi les algorithmes de classification on trouve les algorithmes bio-inspirés, dont font partis les algorithmes de fourmis artificielles. Ces algorithmes se basent sur les techniques de chasse des fourmis ou la manière dont elles trient et organisent leur nid. Nous en distinguons trois catégories :

- Les méthodes d'optimisation basées sur la méta-heuristique *Ant Colony Optimization*. Les algorithmes appartenant à cette catégorie transforment le problème de clustering en problème d'optimisation.
- Les méthodes basées sur l'approche de Lumer et Faieta (LF). La méthode LF s'est elle même inspirée des travaux de Jean-Louis Deneubourg sur les colonies de fourmis. En particulier celui [Deneubourg *et al.*, 1991] traitant de la simulation du comportement d'une colonie de fourmis par des robots pour regrouper un ensemble d'objets identiques dispersés sur une grille. Les règles étant simples : i) si un objet est isolé, elle est prise par un robot ne portant pas de charge, ii) un robot portant une charge dépose l'objet sur le tas ayant plus d'objet. Ce comportement émerge à la fin vers un tas. Par extension, cette catégorie prend en compte d'autres algorithmes de fourmis se basant sur les travaux d'autres biologistes ayant modélisés les comportements de colonies de fourmis.
- Le dernier groupe rassemble des méthodes ayant hybridés les deux autres classes d'algorithmes avec d'autres méthodes. Dans le domaine de la classification cela a été le cas pour l'algorithme AntClass qui combine fourmis artificielles et la méthode des k-moyennes.

## Motivations

Pourquoi et comment positionner une personne handicapée dans un environnement *indoor*? Afin de répondre à cette question nous nous sommes intéressés au :

1. Positionnement des entités. Comme nous l'avons vu, plusieurs technologies peuvent être utilisées et particulièrement celles qui exploitent l'onde électromagnétique. Les réseaux de capteurs sans fils ont été, au début de nos travaux la technologie retenue, afin de collecter les informations de positionnement. Les motivations de ce choix ont été :
  - de savoir si la proximité des réseaux de capteurs de la notion d'intelligence collective nous permettra d'adapter certains des mécanismes provenant des colonies d'êtres vivants sur des problèmes rencontrés dans les réseaux de capteurs tels que le partage des tâches, le routage, la gestion de l'énergie, le clustering<sup>2</sup>.
  - d'apporter une ou des solutions au problème de la localisation par réseaux de capteurs. Pour cela, nous avons entrepris une étude approfondie des méthodes de positionnement. Comme premier constat, il nous est apparu que les réseaux de capteurs ne sont qu'un support pour obtenir les informations de positionnement avec leurs propres contraintes dont la faible consommation d'énergie d'où

---

2. Le clustering consiste, dans les réseaux de capteurs, à construire des *clusters*, c'est-à-dire des groupes de nœuds dans lesquels, un seul nœud est chargé d'exécuter la ou les tâches les plus gourmandes en ressources. Par exemple la transmission des données. Ce nœud est appelé *cluster-head*.

la transmission par la norme Zigbee, entre autres. Ceci nous a poussé à nous intéresser aux autres normes de transmission sans fil : le WiFi, le Bluetooth, le RFID/NFC, le GSM/GPRS.

Une fois les informations fournies par le réseau, on doit appliquer une méthode de positionnement. Ici aussi nous avons envisagé plusieurs méthodes dont une basée sur les fourmis artificielles notamment une méta-heuristique, nommée *Pachycondilla Apicalis*. Cette méthode sera appliquée dans un premier temps sur la classification automatique de données de façon générale et sur le positionnement à l'intérieur des bâtiments après que celui-ci a été converti en problème de classification. Pour ce faire, nous avons mené une étude de la classification automatique de données de façon approfondie. Ceci afin de mieux mesurer les avantages et inconvénients de ces méthodes et de pouvoir appliquer la classification par fourmis au positionnement *indoor*.

2. Le handicap. La position est une information brute qui sert d'entrée à d'autres applications. Il s'agit d'apporter des services basés aussi bien sur la position que sur le handicap afin de faciliter la vie de ces personnes. Ceci va de la navigation dans un bâtiment et les évacuations jusqu'à offrir des services ludiques. Les personnes handicapées se caractérisent par divers niveaux de handicap ayant des contraintes différentes. Nous avons étudié comment les prendre en compte dans un système de positionnement.
3. Les environnements *indoor*. Toutes personnes est amenée à se déplacer dans plusieurs environnements différents. De son domicile à son lieu de travail, dans des endroits publics, dans des lieux de loisirs. Il n'est pas garanti que tous ces environnements disposent des aménagements pour les personnes handicapées, et des technologies de positionnement ou même de réseaux de transmission de données. Et si c'est le cas, est-ce que le déploiement du réseau permet un positionnement ? Est-ce que la compatibilité est garantie entre les technologies des divers environnements ? Aussi, en fonction du handicap qu'elle est la précision requise pour l'application ? Comment déployer les technologies de positionnement afin de répondre à cette exigence ? Et combien de personnes handicapées peuvent être positionnées par le système sans dégrader les performances ? Nous avons proposé une modélisation des environnements *indoor* afin d'effectuer une simulation permettant d'étudier les questions précédentes.

## Organisation des chapitres

Le chapitre 1 présente les réseaux de capteurs d'un point de vue de la localisation. Il met l'accent sur les ondes électromagnétiques, la modélisation et la simulation des réseaux de capteurs. Nous y avons détaillé les normes : WiFi, Bluetooth, IEEE 802.15.4, Zigbee, RFID/NFC. Le chapitre 2 étudie les techniques de mesure et aussi les algorithmes de positionnement. Nous avons fait une distinction entre technologie de localisation, étudiées dans le premier chapitre, et les algorithmes qui ont été étudiés dans ce chapitre. Les systèmes de positionnement ainsi que les systèmes AAL sont étudiés dans le chapitre 3. Nous y présentons les spécificités de la localisation *indoor* des personnes handicapées et

## INTRODUCTION

---

âgées. Le chapitre 4 présente nos propositions et nos résultats pour la localisation *indoor*. Le chapitre 5 présente le problème de la classification automatique de données ainsi que la base des algorithmes de fourmis pour la classification. Dans le chapitre 6, nous présentons la méthode et les expérimentations pour la classification de données. Le dernier chapitre conclut et donne des perspectives.



Première partie

État de l'art de la localisation *indoor*  
par réseaux de capteurs



# Chapitre 1

## Introduction aux réseaux de capteurs

### 1.1 Introduction

Les réseaux de capteurs représentent un domaine de recherche très actif et sont à la base de plusieurs applications. Dans ce chapitre, nous allons étudier ces réseaux dans la perspective de la localisation à l'intérieur des bâtiments. Les réseaux de capteurs peuvent être exploités pour la localisation de plusieurs manières. En effet, déployés dans une zone géographique pour mesurer certains paramètres, le changement de l'un de ces paramètres à un endroit donné peut indiquer la position. Ce qui suppose que ces positions soient déjà connues. Mais c'est surtout la faculté des nœuds à communiquer par ondes radio qui est le plus souvent exploitée. Nous allons montrer les difficultés liées à l'usage des ondes électromagnétiques pour la localisation dans des milieux couverts, comme les bâtiments.

### 1.2 Présentations

Un réseau de capteurs est un ensemble de nœuds qui interagissent les uns avec les autres mais aussi avec le monde extérieur. Il est déployé dans une zone géographique donnée afin de mesurer les grandeurs physiques de celle-ci. Par exemple la température ambiante. Les données ainsi obtenues sont collectionnées et des actions spécifiques peuvent être déclenchées à la suite d'un ou de plusieurs événements.

Les nœuds peuvent être classés en trois types : i) les nœuds sources ; ii) les nœuds destinataires ; iii) les nœuds relais. Sachant qu'un même nœud peut jouer alternativement tous les rôles. Bien que les nœuds puissent être de configuration identique (puissance de calcul, même unité de communication, énergie, stockage), certains d'entre eux ont un rôle particulier. Ainsi dans certaines applications des actionneurs sont utilisés pour exécuter une tâche donnée (appuyer sur un interrupteur, ouvrir une porte, etc.) et d'autres nœuds, appelés puits, servent de ponts entre plusieurs réseaux [Fleury et Simplot-Ryl, 2009].

La communication dans les réseaux de capteurs peut se faire de façon filaire et/ou sans fil. Cependant, c'est la communication sans fil qui est couramment utilisée pour plusieurs raisons dont la facilité de déploiement et la réduction des coûts. La transmission radio est la plus populaire, mais on peut aussi trouver la transmission par infrarouge et par

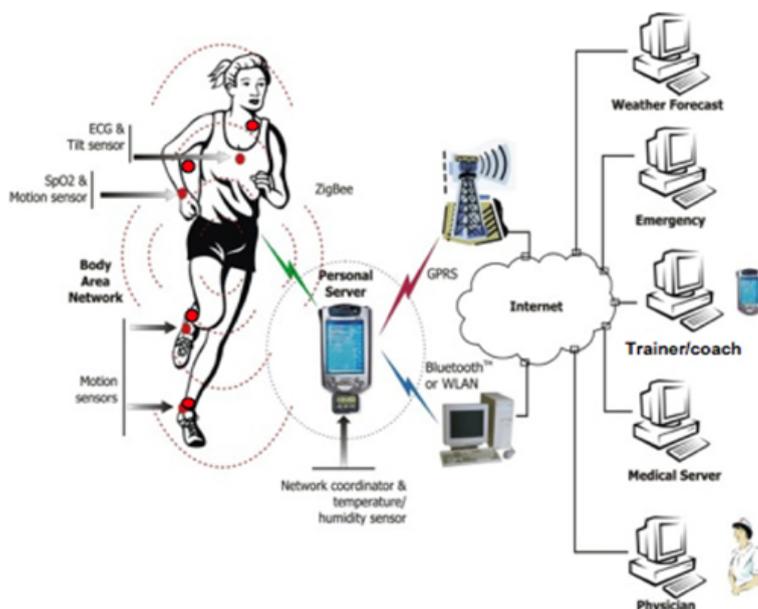


FIGURE 1.1 – Un réseau de capteurs sans fil corporel ou *Wireless Body Sensor Networks*. Il permet de surveiller certaines données physiologiques de la personne et de les envoyer via son PDA à diverses applications. Source [Khaled et Gijs, 2013]

ultrason [Karl et Willig, 2007].

Le concept de la communication machine à machine (*Machine-to Machine* M2M) peut être implémenté dans les réseaux de capteurs ou en les utilisant [Gupta et Hirdesh, 2007]. M2M permet à des nœuds de communiquer et collaborer pour la réalisation d'une tâche sans intervention humaine. Plusieurs des technologies de communication actuelles peuvent être exploitées dans ce but. *Cyber-Physical Systems* [Lee, 2008] (CPS) est un autre concept se basant sur des objets communiquant intelligents.

De plus en plus d'appareils électroménagers, de voitures et d'objets divers disposent de SOCs (*System On Chip*) leur permettant aussi bien de mesurer des grandeurs physiques, de traiter des données et aussi de les transmettre. L'internet des objets (*Internet of Things*) est un autre paradigme qui vise à connecter ces appareils au travers du réseau Internet en utilisant notamment la pile IPv6 [Atzori *et al.*, 2010].

Suivant la taille du domaine de déploiement, les réseaux de capteurs peuvent être classés en : i) des réseaux sur une puce (on-chip network); ii) des réseaux sans fil personnel (Wireless Personal Area Network); iii) des réseaux locaux sans fil (Wireless Area Local Networks) [Gómez *et al.*, 2010, Lewis, 2004].

Comme exemple, la figure 1.1 illustre un réseau de capteurs sans fil corporel (*Wireless Body Sensor Networks*) permettant de surveiller les performances d'un sportif de haut niveau [Khaled et Gijs, 2013]. Ce réseau est constitué de capteurs attachés au corps du sportif. Chaque nœud assure la mesure de ses données physiologiques. Les nœuds pourront pré-traiter les données avant de les envoyer, via une liaison Zigbee au *Personal Device Assistant* (PDA) ou smartphone du sportif. Le PDA agit comme un nœud puits et une

passerelle qui interconnecte le réseau de capteurs à d'autres types de réseaux comme le GPRS (*General Packet Radio Service*) et le WiFi. A travers une liaison Internet, les données peuvent être transmises à des applications diverses, comme par exemple avertir le médecin ou l'entraîneur.

Les réseaux de capteurs ont des caractéristiques spécifiques, que nous allons décrire dans la section suivante.

### 1.3 Caractéristiques

Un réseau de capteurs est en général considéré comme ayant un grand nombre de nœuds. Cependant, ce nombre dépend de l'application et du déploiement [Gómez *et al.*, 2010].

Le déploiement peut se dérouler de plusieurs façons. En fonction de l'application, le déploiement est préconçu, avec dans le cas de la localisation, des nœuds placés à des positions spécifiques [Agrawal, 2011]. Dans d'autres cas, par exemple des applications militaires, les nœuds sont déployés aléatoirement, largués par exemple par avion [Akyildiz *et al.*, 2002]. Dans ces cas, c'est la faculté d'auto-organisation et d'auto-configuration des nœuds qui permettra la mise en œuvre du réseau. Et le déploiement est modélisé par des processus aléatoires, par exemple un processus de Poisson [Fleury et Simplot-Ryl, 2009]. Une phase de pré-déploiement peut exister selon les cas pour redéployer certains nœuds ou déployer des nœuds supplémentaires.

Une fois déployés, tout ou une partie des nœuds peut être mobile. Le réseau doit continuer de fonctionner dans le cas où certains nœuds ne sont pas joignables. Soit parce qu'ils se sont déplacés dans des zones injoignables ou lorsque la liaison est brouillée intentionnellement.

Le nombre de nœuds et les caractéristiques des régions dans lesquelles les réseaux pourraient être déployés requièrent des caractéristiques importantes qui sont : l'auto-organisation (*self-organization*), fonctionnement sans surveillance (*unattended*) quelquefois durant des années, et l'auto-configuration (*self-configuration*) [Akyildiz *et al.*, 2002].

L'ensemble de nœuds assez nombreux, de capacités limitées mais pouvant collaborer pour réaliser une tâche place, sur le plan conceptuel, les réseaux de capteurs dans la catégorie de l'intelligence en essaim (*Swarm Intelligence*). En effet, ce comportement est observable dans plusieurs colonies d'insectes, comme celles de fourmis [Bonabeau *et al.*, 1999]. Ceci est une des raisons qui nous ont encouragés à étudier les algorithmes de fourmis artificielles pour proposer une solution au problème de la localisation interne par réseau de capteurs se basant sur ce comportement.

La gestion de l'énergie et son corollaire de maximisation de la durée de vie du réseau constituent une autre spécificité des réseaux de capteurs. Contrairement aux réseaux locaux, dont l'infrastructure peut être alimentée par le réseau électrique, les réseaux de capteurs sont conçus en général pour fonctionner sans infrastructure et sur batterie. Cependant, si les capteurs sont alimentés par une source d'énergie pérenne le problème se pose autrement. Mais, il peut arriver que les deux cas coexistent, où certains nœuds sont alimentés par batterie et d'autres par le secteur. Il s'agit dans ces deux derniers cas de maximiser la durée de vie du réseau. Ainsi maximiser la durée de vie passe aussi bien par

la minimisation de l'énergie globale consommée que par le maintien de la connectivité du réseau dans le temps. Ce dernier est obtenu par la répartition de la consommation d'énergie sur l'ensemble du réseau pour limiter le nombre de nœuds qui disparaissent progressivement [Fleury et Simplot-Ryl, 2009].

Du fait des capacités limitées et des contraintes liées à l'énergie, la communication dans les réseaux de capteurs se fait généralement par transmission multi-saut (*multihop*) et est caractérisée par des débits de transmission faibles. La transmission multi-saut contrairement à la transmission un-saut, n'envoie pas directement les données au nœud destinataire, mais utilise les nœuds voisins de proche en proche ou le nœud puits du groupe (*cluster-head*) pour atteindre le nœud destinataire. Le *cluster-head* est élu pour assurer le rôle de nœud puits dans un voisinage donné, pour certaines stratégies de gestion d'énergie ou pour certaines applications (le routage). Celui-ci dispose à un instant donné de plus de ressources pour assurer la tâche [Abbasi et Younis, 2007].

Nous allons maintenant décrire les capteurs constituant le réseau.

## 1.4 Architecture d'un capteur

Un capteur (*smart sensor* ou *sensor node*) désigne un nœud d'un réseau de capteurs. C'est par abus de langage que le mot « capteur » désigne le nœud du réseau qui comporte plusieurs « *dispositifs qui soumis à l'action d'un mesurande non électrique présentent une caractéristique de nature électrique (charge, tension, courant ou impédance) désignée pas s et qui est fonction du mesurande.*

$$s = F(m)$$

» [Asch, 2010]

Où  $s$  est la grandeur de sortie ou réponse du capteur et  $m$  est la grandeur d'entrée ou excitation. Nous avons conservé dans la suite de ce document cette ambiguïté.

Un nœud est muni d'une ou de plusieurs unités lui permettant de collecter des données sur l'environnement, de leur appliquer des prétraitements si nécessaire et de les envoyer au travers une liaison sans fil vers un autre capteur ou un nœud spécifique disposant de plus de puissance de calcul et de stockage, appelé nœud puits (*sink node*). Il joue de ce fait un rôle d'interface entre le monde réel et le domaine d'application. Le Tableau 1.2 illustre certaines de ces grandeurs et les capteurs utilisés pour les mesurer.

Les sous-systèmes ou unités d'un capteur sont :

- l'unité de traitement ;
- l'unité de communication ;
- l'unité de mesure ;
- et accessoirement d'un générateur d'énergie et/ou d'un ensemble d'actionneurs.

L'interconnexion de ces divers unités se fait généralement par des bus série comme les Serial Peripheral Interface (SPI), General Purpose Input/Output (GPIO), Inter-Integrated Circuit ( $I^2C$ ), entre autres [Dargie et Poellabauer, 2011]. Une description de cette architecture est illustrée par la figure 1.2.

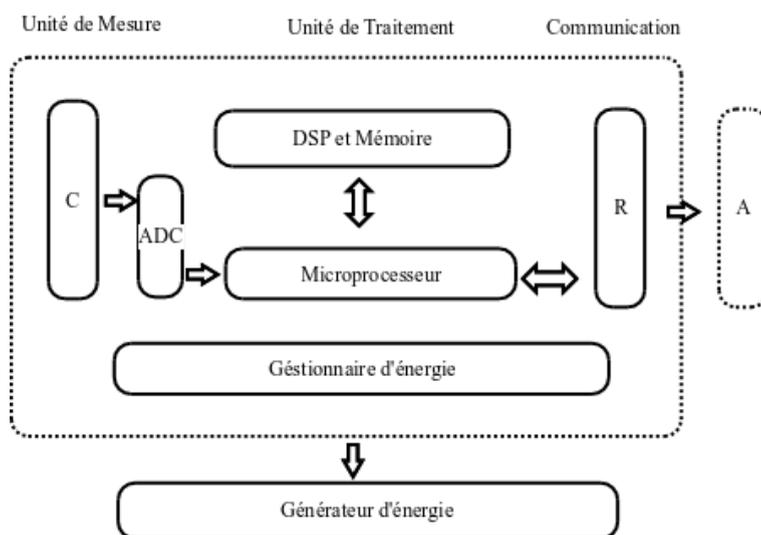


FIGURE 1.2 – Architecture typique d'un capteur. C : unité de mesure pouvant comporter plusieurs capteurs. R : unité de communication. Le R symbolise la communication radio bien que d'autres parties du spectre électromagnétique peuvent être exploitées. Les blocs DSP (*Digital Signal Processing*) Mémoire et Microprocesseur représentent l'unité de traitement. A : les extensions du nœud par des actionneurs permettent de réaliser des tâches spécifiques. Gestionnaire d'énergie permet d'incorporer une méthode particulière de gestion de l'énergie et le générateur d'énergie est une source d'énergie comme par exemple un panneau solaire.

## 1.4. ARCHITECTURE D'UN CAPTEUR

Nom	Fabricant	Caractéristiques	
		UT	Mémoire
	Texas Instruments	MCU (MSP430F123) : 8 MHz	Flash : 8 KB SRAM : 256 octets
	ATMEL	MCU (ATMEGA128L) : 8 MHz	Flash : 128 KB SRAM : 4 KB

TABLE 1.1 – Ce tableau illustre quelques unités de traitement. MCU-MSP et MCU-AVR sont des microcontrôleurs de Texas Instruments et de Avrora.

### 1.4.1 L'unité de traitement

L'unité de traitement exécute les programmes et gère les autres unités. Elle se compose entre autres d'unités de calcul, d'une mémoire non-volatile (mémoire flash) pour stocker les programmes, une mémoire vive et une horloge interne. Un nœud peut être muni d'un ou de plusieurs types d'unités de calcul [Dargie et Poellabauer, 2011] : les microcontrôleurs (MCU), les processeurs de signal numérique ou *Digital Signal Processing* (DSP), les *Application-Specific-integrated-Circuits* (ASICs) et les *Field Programmable Gate Array* (FPGA). Les avancées dans la miniaturisation des circuits électroniques permettent d'avoir des nœuds munis de processeurs plus puissants par rapport à des capteurs basiques. En témoignent la généralisation des processeurs à architecture ARM (Acorn Risk Machine) et des SOCs (System On Chip) dans les téléphones portables intelligents, les tablettes, etc. La Table 1.1 illustre quelques unités de traitement.

Le coût, la taille, la consommation d'énergie et les performances, le fait que l'unité de traitement soit spécifique à une application ou générale, constituent des contraintes pour le choix du nœud [Dargie et Poellabauer, 2011].

### 1.4.2 Unité de mesure

Cette unité intègre un ou plusieurs capteurs faisant l'interface avec le monde réel ainsi que plusieurs convertisseurs Analogique-Numérique et des multiplexeurs. Un exemple est donné par le tableau 1.2.

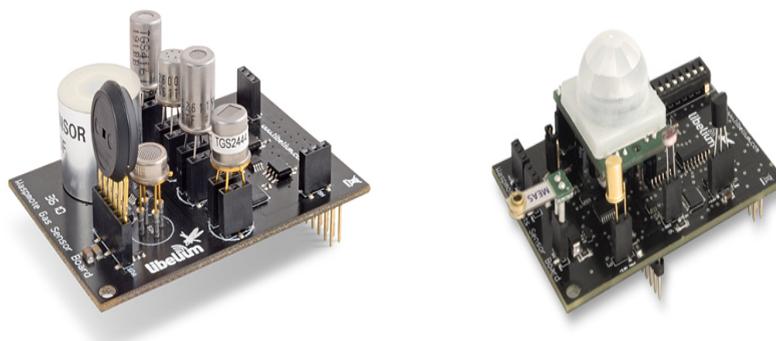
Ces capteurs sont utilisés dans plusieurs applications, surtout celles liées à l'informatique ubiquitaire (*ubiquitous computing*) et les applications sensibles aux contextes (*context-aware*) qui extraient des connaissances des données provenant de la multitude de capteurs (température, luminosité, etc.) et fournissent des services aux utilisateurs.

Le déploiement de capteurs dans un environnement permet de rendre celui-ci intelligent (*smart*) et est à la base des concepts tels que l'intelligence ambiante (*Ambient Intelligence*), les villes intelligentes (*smart-cities*) et d'autres concepts similaires (*smart-car*, *smart-transport*, etc.). La figure 1.3 illustre quelques unités de mesures du nœud Waspnote.

## 1.4. ARCHITECTURE D'UN CAPTEUR

Grandeur physique	Exemple de Capteur	Exemple d'application
Température	Thermostat, Thermocouple	Domotique
Humidité	capteurs résistive et capacitive, hygromètre	Domotique
Lumière	Capteur infrarouge, Photodiode	Domotique
Son	Microphone, résonance piézoélectrique	Localisation
Mouvement	Accéléromètre, gyroscope	Localisation
Position	capteurs de présence : ultrason, infrarouge	Localisation
Champ Électromagnétique	magnétomètre	Localisation

TABLE 1.2 – Description de quelque grandeurs physiques et de certains capteurs utilisés pour les mesurer.



a) Module de mesures des Gaz    b) Module de mesure d'événements

FIGURE 1.3 – a) Modules pour mesurer les émanations dans les villes et d'autres zones. Il permet par exemple de mesurer le niveau de dioxyde de carbone, d'Oxygène, etc. b) Ce module permet de détecter la vibration, les effets de hall (portes et fenêtres) et la détection de personne PIR entre autres. Source [Libelium, 2012].

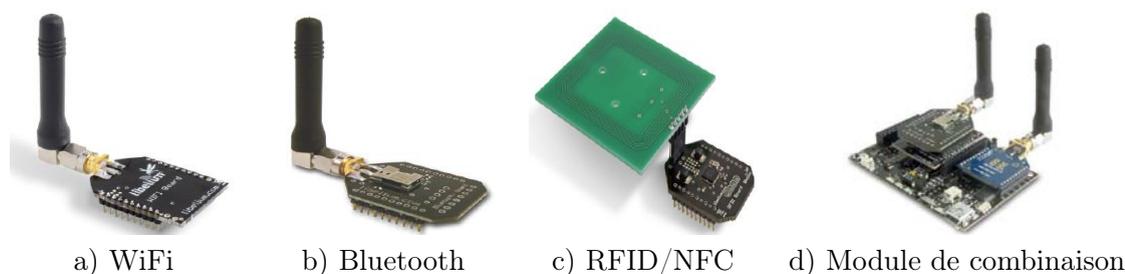


FIGURE 1.4 – Illustration des quelques modules utilisés par le nœud Wapsmote. Source [Libelium, 2012]

Les capteurs utilisent un transducteur qui permet de convertir une grandeur physique en une autre, généralement en courant électrique. Ces grandeurs étant analogiques, il est de ce fait nécessaire d'utiliser un convertisseur Analogique-Numérique (ADC). Le processus de conversion se caractérise par des pertes. Un bon capteur étant celui qui mesure avec une grande précision un événement survenu [Dargie et Poellabauer, 2011].

### 1.4.3 L'unité de communication

L'unité de communication transmet et reçoit des informations des nœuds environnants. Un nœud peut être muni de plusieurs modules de communication filaire et/ou sans-fil avec des débits qui sont en général faibles. Cependant, du fait de la grande consommation d'énergie induite par ces modules, seuls certains nœuds peuvent être munis de plusieurs modules de communication. La Figure 1.4 illustre quelques modules de communication utilisables par un nœud Wapsmote. Elle montre aussi un module d'extension permettant une combinaison des technologies de transmission, par exemple WiFi/Zigbee, GSM/NFC, etc.

Pour la transmission radio, ce sont les bandes de fréquences Industrie-Science-Médecine (ISM) qui sont les plus utilisées pour la raison que ces bandes ne sont pas soumises à des régulations [Akyildiz *et al.*, 2002]. Mais du fait du grand nombre d'applications qui utilisent ISM, les interférences sont nombreuses. Par exemple aussi bien les normes WiFi, Zigbee et le Bluetooth utilisent la bande des 2.4 GHz. Des solutions pour éviter les interférences, résident dans la façon dont chacune de ces normes divise la bande en canaux, des modulations utilisées, etc.

L'Ultra Large Bande représente une alternative aussi bien par ce que la transmission par ULB n'utilise pas les mêmes bandes de fréquences et aussi la localisation par ULB est plus précise en utilisant le temps d'arrivée du signal. Nous allons détailler la communication dans la section 1.5.

### 1.4.4 Extension de la définition d'un capteur

Pour nos travaux, nous avons élargi la définition d'un capteur pour prendre en compte tout appareil disponible dans la zone de déploiement du système de localisation et pouvant émettre et recevoir, et muni d'une unité de traitement lui permettant d'exécuter les pro-

grammes, de stocker les résultats avant de les envoyer à un autre nœud ou un nœud puits. Des exemples d'appareils à usage courant sont le réfrigérateur, le téléviseur, la caméra et bien entendu le téléphone portable devenu un enjeu pour plusieurs applications. Le concept du « *smart-everything* », s'étend de plus en plus à d'autres appareils dont nous ne pouvons pas donner une liste exhaustive.

Pour prendre en compte cette hétérogénéité, nous avons proposé une architecture en trois couches [Amadou Kountché *et al.*, 2011] : i) une couche réseau générant la création des réseaux spécifiques ( WiFi, Zigbee, etc) ; ii) un middleware permettant l'interopérabilité entre ces divers nœuds et la transparence pour les utilisateurs ; iii) une couche application.

Pour un nœud, l'énergie est une contrainte importante. Avec cette extension nous devons considérer cette diversité dans la zone de déploiement par exemple, le domicile d'une personne handicapée et pouvoir fournir des méthodes globales d'économie d'énergie pour que les appareils fonctionnant sur batterie et ceux sur le secteur puissent tout le temps former un réseaux connecté.

### 1.5 La communication dans les réseaux de capteurs

Comme, nous l'avons déjà mentionné, la communication dans les réseaux de capteurs est contrainte par l'énergie. L'usage de certaines technologies, comme le WiFi, qui n'ont pas été spécifiées à l'origine pour des réseaux de faible capacité, nécessitent la prise en compte de cette contrainte dans la conception des systèmes de localisation.

Un signal électromagnétique  $s(t) = a \sin(2\pi ft + \phi)$ , d'amplitude  $a$ , de fréquence  $f$  et de phase  $\phi$ , en se propageant dans un milieu donné subit des atténuations modélisées par une fonction  $h$ . Cette fonction est composée d'une composante déterministe, nommée affaiblissement et de deux composantes stochastiques : les effets de masque et les évanouissements non sélectifs. La modélisation de  $h$  est donnée dans la section 1.6.

Dans les meilleures conditions, *ie* en l'absence de perturbations dans le milieu de propagation, le signal est reçu en ligne de propagation directe ou *Line of Sight* (LOS). La puissance reçue, le temps d'arrivée du signal, sa phase ou son angle d'arrivée peuvent servir à estimer la distance séparant l'émetteur et le récepteur. Cependant dans la réalité, plusieurs facteurs, que nous allons décrire dans la section 1.6, compliquent ce cas simple.

Dans cette section nous allons présenter les technologies de transmission sans fil les plus utilisées pour la localisation à l'intérieur des bâtiments.

#### 1.5.1 La norme IEEE 802.11 (ISO/IEC 8802-11)

IEEE 802.11 (802.11x)<sup>1</sup> est un ensemble de protocoles se basant sur le même protocole de gestion du médium d'accès (MAC) et de la couche physique (PHY) qui a été la base de plusieurs autres protocoles comme le Bluetooth et le Zigbee [Fleury et Simplot-Ryl, 2009]. C'est le travail de Bahl [Bahl et Padmanabhan, 2000] qui a aussi bien popularisé l'usage du 802.11 que du fingerprinting pour la localisation à l'intérieur de bâtiment. Les systèmes de

---

1. Cette norme est aussi appelée par abus de langage WiFi. Wireless Fidelity (Wi-Fi) est une certification délivrée par la WiFi-Alliance aux matériels conforme à la norme IEEE 802.11.

localisation commercialisés par Ekahau [Ekahau, 2013] et par Qubulus [Qubulus, 2013b] et les efforts de la société Nokia à travers l’initiation du *Indoor Location Alliance* [Nokia, 2012] témoignent des efforts pour la commercialisation à grande échelle et à des coûts accessibles de systèmes de localisation par WiFi. Les raisons de l’engouement pour le positionnement par WiFi sont nombreuses dont : la disponibilité de l’infrastructure de ces réseaux, l’amélioration constante de cette norme et la réduction des coûts de déploiement et de maintenance. Le positionnement par WiFi est aussi relativement facile à mettre en œuvre car ne nécessitant pas le déploiement de nœuds supplémentaires et coûteux. Bien que la précision soit métrique, elle suffit pour la plupart des applications [Mautz, 2012].

Le tableau 1.3 illustre les deux couches définies par la norme 802.11.

Couches OSI	Couches IEEE	Normes
Liaison	LLC (Logical Link Control) MAC (Medium Access Control)	802.2 802.11
Physique	PhY (Physical Signaling Layer)	IR FHSS DSSS OFDM

TABLE 1.3 – Couches des protocoles IEEE 802.11

Les trames de la couche liaison de la norme 802.11 peuvent être transmises au travers de quatre couches physiques qui sont [Lohier et Présent, 2010] :

- IR (infrarouge) est moins utilisé pour la transmission de données sur les réseaux locaux. Cependant, il peut être exploité pour une localisation à l’intérieur de bâtiment par positionnement par proximité ;
- FHSS (Frequency Hopping Spread Spectrum) est une technique d’émission par étalement de spectre qui consiste à émettre sur toute la largeur de la bande avec un saut de fréquence (un changement de canal). Le saut de fréquence est effectué suivant une séquence pseudo-aléatoire connue de l’émetteur et du récepteur. Dans le cas du IEEE 802.11, la bande des 2,4 (2,4000 - 2,4835 Ghz) est divisée en 79 canaux de 1 MHz chacun et le saut est effectué toutes les 20 ms suivant la même séquence à l’intérieur de la même cellule. Cette technique accroît l’immunité au bruit, est robuste face aux interférences et permet un meilleur partage de la bande de fréquence entre plusieurs transmissions différentes [McCune, 2000]. Le débit est limité à 2 Mbit/s ;
- DSSS (Direct Sequence Spread Spectrum) est une technique d’étalement de spectre qui combine le signal à émettre avec un signal pseudo-aléatoire de bande de fréquence plus élevée. La technique de la séquence directe divise la bande des 2,4 GHz en 14 canaux de 20 MHz chacun. La transmission ne se fait que sur un seul canal par cellule pour un réseau IEEE 802.11. Les avantages du DSSS sont la transmission numérique, résistance au brouillage et aussi le partage de la bande de fréquences ;
- OFDM (Orthogonal Frequency Division Multiplexing). Cette technique permet de répéter le signal à transmettre sur différentes fréquences porteuses orthogonales dans le but d’augmenter le débit global. Suivant la norme associée, 802.11a ou 802.11g, le multiplexage OFDM utilise respectivement la bande U-NII (Unlicensed-National Information Infrastructure) dans les 5 GHz ou la bande ISM des 2,4 GHz.

Plusieurs révisions ont été apportées à la norme IEEE 802.11 aussi bien pour assurer une meilleure sécurité qu’une meilleure interopérabilité. Les versions les plus importantes de la norme 802.11, dites normes 802.11 physiques, sont données dans le tableau 1.4.

## 1.5. LA COMMUNICATION DANS LES RÉSEAUX DE CAPTEURS

Version	Fréquence	Débits
802.11 (Legacy)	2,4 - 2,5 GHz	1 - 2 Mbit/s
802.11a	5 GHz	25 - 54 Mbit/s
802.11b	2.4-2.5 GHz	6.5 - 11 Mbit/s
802.11g	2.4-2.5 GHz	25 - 54 Mbit/s
802.11n	2.4 GHz ou 5 GHz	200 - 500 Mbit/s

TABLE 1.4 – Tableau récapitulatif des versions importantes de la norme IEEE 802.11

### Architecture du IEEE 802.11

Un réseau 802.11 est constitué de deux types d'équipements [Lohier et Présent, 2010] : i) les stations sans fils (tout appareil équipé d'une carte WNIC (*Wireless Network Identification Card*) et ii) les points d'accès (AP, *Access Point*) qui permettent la coordination des transmissions et servent de pont entre réseaux câblés et sans fils. Chaque appareil est identifié par son adresse physique dite adresse MAC sur 48 bits.

Le réseau est basé sur une topologie cellulaire dont chaque cellule (*Basic Service Set*, BSS) est gérée par un point d'accès. Deux modes de fonctionnement peuvent être choisis : le mode infrastructure (*Infrastructure Mode*) et le mode *ad hoc* ou mode sans infrastructure.

En mode infrastructure, les points d'accès sont interconnectés par une dorsale (*backbone*) câblée appelée système de distribution (*Distribution System*, DS). L'interconnexion des cellules s'appelle ensemble étendu de cellules (ESS, *Extended Service Set*). Lorsque le mode *ad hoc* est choisi, les stations communiquent directement entre elles sans passer par une infrastructure. Les cellules sont appelées des IBSS (*Independent Basic Service Set*). Ces deux modes de fonctionnement sont illustrés sur la Figure 1.5.

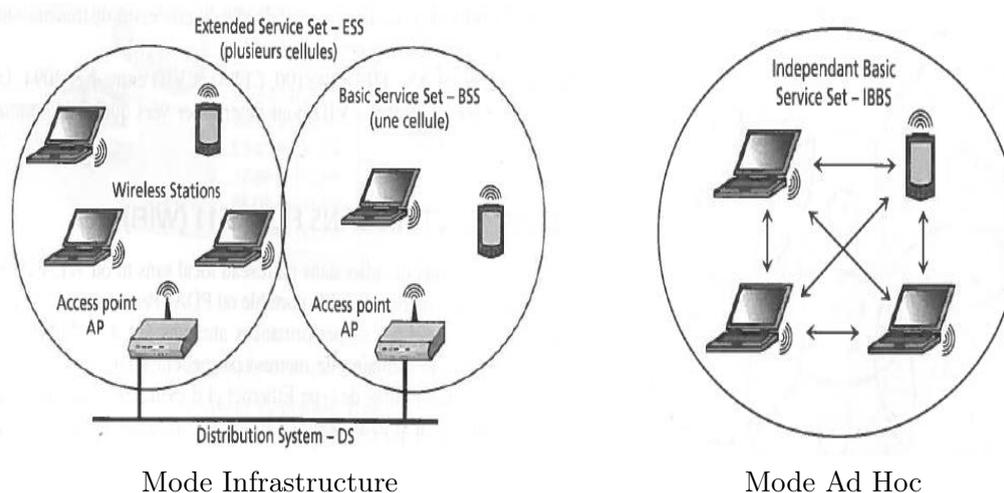


FIGURE 1.5 – Les deux modes de fonctionnement d'un réseau WiFi. Source[Lohier et Présent, 2010]

### Association et Handover

L'association est le processus durant lequel une station cherche à s'associer à une cellule (accéder à un BSS ou à un IBSS) après un démarrage, un passage en mode veille ou un déplacement. Le nœud communique avec les APs, ou les autres nœuds dans le cas *ad hoc*, afin d'obtenir des informations de synchronisation.

L'échange peut se dérouler de deux façons en fonction de la puissance des signaux reçus ou de la consommation d'énergie occasionnée par les transmissions [Lohier et Présent, 2010] :

- écoute passive. Ce sont les APs qui émettent périodiquement des trames balises contenant les informations de synchronisation et aussi des informations sur l'AP. Les stations se synchronisent en recevant ces trames.
- écoute active. C'est la station qui essaie de rejoindre un AP en transmettant une trame de requêtes (*Probe Request Frame*) et attend la réponse du point d'accès.

L'association contient un processus d'authentification qui peut être sécurisé (WEP, WAP, etc.). Un échange d'informations concernant les caractéristiques de la cellule clôt l'association et la station peut transmettre et recevoir des trames de données. L'émission périodique de trames balises assure la synchronisation tout au long des échanges et aussi permet à un point d'accès de signaler sa présence. La fréquence d'émission des trames est paramétrable sur chaque point d'accès.

Le handover est le processus qui maintient la communication d'une station en déplacement d'une cellule vers une autre sans interruption. La station teste la puissance du signal de l'AP et si celui-ci s'affaiblit, elle se ré-associe avec un nouvel AP sur un autre canal. Lors d'un déplacement en mode *ad hoc*, les stations testent, de la même façon, la présence d'autres stations sur différents canaux.

### Localisation par IEEE 802.11

Ces deux processus (association et handover) peuvent être exploités pour la localisation à l'intérieur de bâtiment. Dans le cas de l'association, un positionnement peut être obtenu à travers les deux méthodes d'association. Ainsi, une station peut extraire le temps d'arrivée de la trame balise ou la puissance et estimer la distance. Le nœud mobile peut ainsi appliquer une méthode de triangulation pour estimer sa position après réception d'au moins trois balises de trois AP. Néanmoins ce type de positionnement requière pour l'usage du TOA une modification des trames pour contenir le temps d'émission et aussi une synchronisation entre les stations. Et en pratique, les modèles théoriques distance-puissance ou distance-temps ne sont pas faciles à appliquer car les API des pilotes des cartes WiFi ne fournissent pas toujours ces informations et les cartes ne sont pas conçues dans cette perspective. Cependant, plusieurs constructeurs (Cisco, TRX, Broadcom, Nokia, Qualcomm, etc.) [Qubulus, 2013a] offrent des matériels pour un tel positionnement.

Au moment d'un handover, si le déploiement des AP est soigneusement choisi, la position d'une station peut être déterminée comme celle du AP auquel il vient de s'associer. Ce positionnement approximatif n'offre cependant pas une grande précision mais est simple à mettre en œuvre.

La localisation par WiFi se base principalement sur la détermination de l'empreinte de la propagation du signal dans l'environnement *indoor*. Cette empreinte est construite

par la mesure des puissances du signal à un point donné et provenant de plusieurs points d'accès. Soient  $(x, y)$  le point en question, et  $RSS_i(x, y)$  la puissance du signal provenant du point d'accès  $i$ . L'ensemble des mesures pour un point donné est :

$$RSS(x, y) = [rssi_1(x, y), \dots, rssi_n(x, y)]$$

Avec  $n$  le nombre de points d'accès. Toutes ces mesures sont conservées dans une base de données. Une fois les mesures obtenues, la position est déterminée par des algorithmes de reconnaissance des formes comme les réseaux de neurones, les méthodes des plus proches voisins, etc.

Cette phase de mesure est coûteuse en temps car il faut se déplacer dans le bâtiment à plusieurs points et à divers moments. Et il est nécessaire d'effectuer plusieurs mesures pour un même point afin de réduire les effets du bruit. Deux approches sont utilisées à cette fin. La première consiste à déterminer la moyenne des  $n$  mesures ou un estimateur plus robuste aux valeurs aberrantes comme la médiane. La seconde approche est de déterminer la distribution de probabilité de la puissance du signal à partir des mesures, ce qui est effectué par l'acquisition de plusieurs mesures. Les distributions de probabilités sont alors utilisées dans des méthodes probabilistes telles que *maximum a priori* (MAP) ou des méthodes d'inférence bayésienne pour déterminer la position.

Notons que la phase de mesure peut être simplifiée par la collaboration entre les utilisateurs pour améliorer le positionnement à l'image de Wikipedia [Bolliger, 2008].

### 1.5.2 La norme IEEE 802.15 (Bluetooth)

Le Bluetooth est une technologie qui s'adresse aux réseaux personnels sans fil ou WPAN (*Wireless Personal Area Network*). Il se caractérise par une courte portée radio, une simplicité d'intégration et des faibles ressources. Le Bluetooth est actuellement utilisé pour connecter plusieurs équipements différents comme des casques sans fil à un téléphone, le transfert de données, etc. Cette norme a connu plusieurs améliorations qui se sont traduites en version : 1.0, 1.1, 1.2, 2.0, 2.0 + EDR, 2.1 + EDR, 3.0 + HS, 4.0 [Bluetooth, SIG, 2013]. Comme pour le WiFi, la Table 1.5 montre les couches définies par la norme IEEE 802.15.

Couches OSI	Couches IEEE	Normes
Liaison	LLC (Logical Link Control)	802.2
	MAC (Medium Access Control)	802.15
Physique	PhY (Physical Signaling Layer)	FHSS + GFSK UWB

TABLE 1.5 – Modèle IEEE 802.15. Source[Lohier et Présent, 2010]

La couche liaison du 802.15 contrairement à celle du 802.11 utilise une version du Time Division Multiple Access (TDMA) pour allouer le canal radio. Ainsi l'allocation du canal est simplifiée par l'architecture maître-esclave où c'est le maître qui alloue des temps d'émission à ses esclaves. Le maître émet dans les intervalles pairs et les esclaves dans les intervalles impairs.

La couche physique du Bluetooth a d'abord utilisé la bande de fréquence 2,4 GHz. Cette bande est divisée en 79 canaux de 1 MHz chacun et une modulation GFSK est utilisée pour

Classe	Puissance	Portée
1	100 mW (20 dBm)	100 mètres
2	2,5 mW (4 dBm)	10 à 20 mètres
3	1 mW (0 dBm)	Quelques mètres

TABLE 1.6 – Les classes d’appareils Bluetooth.

offrir un débit de 1 Mbits/s. Cette première version est appelée Bluetooth BR (*Basic Rate*). Et peut être combinée avec le EDR.

*Enhanced Data Rate* (EDR) est une amélioration qui permet d’utiliser la bande 2,4 Ghz avec une modulation  $\frac{\pi}{4}$ -DQPSK et 8-DPSK pour offrir des débits de respectivement 2 à 3 Mbits/s entre des appareils compatibles EDR.

À partir de la version 3.0 une autre amélioration appelée *High Speed* (HS) permet d’utiliser le Bluetooth sur une couche IEEE 802.11 pour offrir un débit théorique de 24 Mbit/s. Cette modification n’est supportée que par les matériels labellisés HS. Il faut néanmoins rappeler que le Bluetooth vise principalement des applications bas débits et de courte portée. L’utilisation du haut débit, par rapport au WiFi, permet une simplification des transmissions.

Le protocole Bluetooth est actuellement dans sa quatrième version qui apporte plusieurs améliorations dont le Bluetooth Low Energy (BLE), une meilleure sécurité par cryptage AES. Il assure aussi la rétrocompatibilité avec la version classique et la version HS. BLE est une nouvelle sous-couche de protocole de la version 4.0 destinée à une rapide mise en place de liaisons Bluetooth simples avec pour objectif la minimisation de l’énergie consommée [Bluetooth, SIG, 2013].

### Architecture d’un réseau IEEE 802.15

Un réseau Bluetooth est un réseau *ad hoc* organisé autour d’un ou de plusieurs maîtres ayant chacun au maximum sept esclaves en mode actif. Chaque nœud dispose d’une adresse MAC et peut aussi être caractérisé par sa puissance d’émission résumé dans le tableau 1.6. Cependant la plupart des *transceivers* Bluetooth sont de classes 2.

Un maître et ses esclaves forment un « picoréseau » (*piconet*). Dans un piconet, c’est le maître qui initie la liaison ; les nœuds esclaves sont tous synchronisés par le maître. Puisque la topologie du réseau est en étoile, toutes les communications passent par le maître.

Plusieurs « picoréseaux » peuvent être interconnectés pour former un *scatternet*. Dans ce dernier, un nœud peut être maître dans un *piconet* et esclave dans un autre. Il est aussi possible pour un nœud de changer de rôle. Par exemple un nœud initie une liaison Bluetooth en tant que maître et une fois la liaison établie, il se transforme en esclave.

Dans un réseau Bluetooth, c’est le maître qui initie la communication et gère le réseau.

### Localisation par Bluetooth

Comme nous l’avons vu pour le WiFi, le positionnement peut être implémenté en utilisant cette norme en se basant sur le temps, l’angle, la puissance du signal reçu.

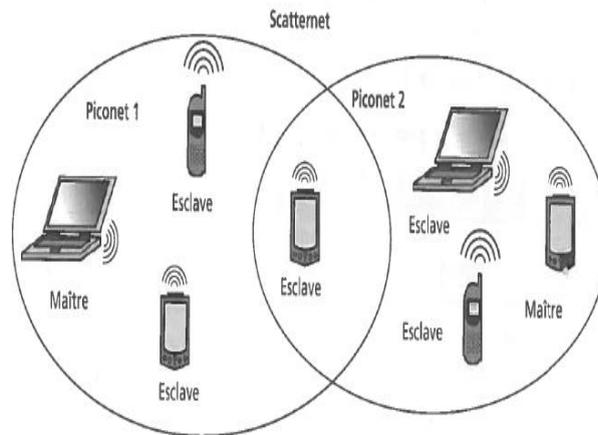


FIGURE 1.6 – Architecture d'un réseau Bluetooth. Source [Lohier et Présent, 2010]

Le fingerprinting [Feldmann *et al.*, 2003, Genco, 2004, Pei *et al.*, 2010] est aussi exploitable pour le positionnement par Bluetooth. Bekkelien [Bekkelien *et al.*, 2012] a mis en place un système de positionnement par Bluetooth en utilisant un algorithme des  $k$  plus proches voisins et une méthode bayésienne naïve.

Les avantages du positionnement par Bluetooth sont la simplicité de mise en œuvre grâce aux faibles coûts d'acquisition des transceivers Bluetooth et l'architecture maître esclave. Cette architecture [Cheung *et al.*, 2006] favorise un positionnement simple par proximité. En effet, un déploiement adapté aux besoins de l'application visée permet d'apporter une précision de niveau chambre ou sous-chambre en s'aidant de la portée des nœuds, par exemple la classe 3 n'a que quelques mètres de portée.

L'inconvénient des versions classiques du Bluetooth, comme du WiFi, sont la consommation d'énergie. Pour un smartphone, par exemple, le niveau de la batterie fond littéralement lorsque le Bluetooth, le WiFi et le GPS sont activés. Ce à quoi, veut répondre le BLE et aussi la norme IEEE 802.15.4. Une analyse des propriétés du signal Bluetooth pour le positionnement a été effectuée dans [Madhavapeddy et Tse, 2005, Nguyen et Luo, 2013].

### 1.5.3 La norme IEEE 802.15.4

La norme IEEE 802.15.4 spécifie une couche physique et une couche de contrôle d'accès au médium pour les réseaux sans fils à faible débit ou LR-WPAN (Low Rate WPAN).

Un réseau personnel sans fils à faible débit est défini [IEEE Computer Society, 2005], par le comité 802.15, comme « *un réseau de communication simple et à faible coût qui autorise une connexion sans fils pour les applications ayant besoin de puissance limitée et des contraintes relâchées sur le débit. Les objectifs principaux d'un LR-WPAN sont la facilité d'installation, des transferts de données fiables, des coûts très faibles et des durées de vie raisonnable pour les batteries tout en fournissant un protocole simple et flexible.* »

Comme pour les deux dernières normes, le tableau 1.7 montre le positionnement du IEEE 802.15.4 dans le modèle OSI.

Couches OSI	Couches IEEE	Normes
Liaison	LLC (Logical Link Control MAC (Medium Access Control)	802.2 802.15.4
Physique	PhY (Physical Signaling Layer)	DSSS UWB

TABLE 1.7 – Modèle IEEE 802.15.4

La couche MAC fournit 2 façons d'accéder au canal [Lohier et Présent, 2010] : le *Carrier Sense Multiple Acces/Collision Avoidance* CSMA/CA et ALOHA. La couche physique utilise elle aussi plusieurs bandes de fréquences en fonction des zones géographiques :

- Europe. 868.0 - 868.6 MHz . 1 canal. 20/100/250 Kb/s
- Amérique du Nord. 902 - 928 MHz. De 10 à 16 canaux. 40/250 Kb/s
- Monde entier. 2400 - 2483.5 MHz : Jusqu'à 16 canaux. 250 Kb/s

La couche physique est responsable des tâches suivantes [IEEE Computer Society, 2005] :

- Activation et désactivation du transceiver radio ;
- La détection de l'énergie dans le canal courant ;
- L'indication du LQI (Link Quality Indicator) pour les paquets reçus ;
- Sélection de la fréquence du canal ;
- Transmission et la réception des données ;
- Mesure précise pour la couche PHY utilisant l'UWB.

Les couches physiques définies dans ce standard sont :

- O-QPSK : C'est une couche PHY utilisant du DSSS et une modulation OQPSK opérant dans les bandes de fréquences 868 MHz, 915 MHz et 2450 MHz ;
- BPSK : DSSS et une modulation BPSK dans les trois bandes de fréquences ;
- ASK : Du PSSS (parallel sequence spread spectrum) utilisant une modulation amplitude shift keying (ASK) et BPSK opérant dans les trois bandes de fréquences ;
- CSS : Chirp Spead Spectrum utilisant une modulation DPSK (differential quadrature phase-shift keying) opérant dans la bande 2,4 GHz ;
- UWB : Combinaison du burst position modulation (BPM) et du BPSK opérant dans la bande de fréquence 3-10 GHz ;
- MPSK : M-ary phase-shift keying (MPSK) opérant dans les 780 MHz ;
- GFSK : Gaussain frequency shift keying opérant dans les 950 MHz.

IEEE 802.15.4 est la base de plusieurs piles de protocoles dont [Simek *et al.*, 2013] : Zigbee, WirelessHart, ISA100.11a, MiWi et dans une certaine mesure la 6LoWPAN.

### Architecture d'un réseau IEEE 802.15.4

Un réseau IEEE 802.15.4 se compose de deux types d'équipements : un *full-function device* (FFD) et un *reduced-function device* (RFD). Il peut fonctionner, selon les besoins de l'application, en topologie étoile, pair-à-pair ou en arborescence. Tous les équipements fonctionnant dans le réseau, dans les deux topologies, ont une adresse unique appelée adresse étendue sur 64 bits.

Un FFD est un équipement disposant de plus de ressources (puissance de calcul, capacité mémoire, énergie, etc. ) lui permettant de jouer le rôle de coordinateur du réseau personnel ou de routeur. Le coordinateur est un contrôleur central qui a pour rôle la créa-

tion du réseau et l'association de nouveaux nœuds au réseau. Cependant pour assurer le rôle de coordinateur, peu de ressources sont requises mais un réseau IEEE 802.15.4 doit avoir un et un seul coordinateur.

Un RFD a peu de ressources et est la plupart du temps en sommeil pour économiser l'énergie. Il sert selon les applications d'actionneur, de capteurs, etc. Un réseau IEEE 802.15.4 dispose de plusieurs nœuds de ce type.

La communication par topologie étoile est établie entre les autres nœuds et le coordinateur du PAN. Les nœuds ne peuvent pas communiquer directement entre eux car toutes les communications doivent passer par le coordinateur. Un nœud est typiquement associé à une application et est soit le point de départ ou de terminaison des communications réseaux.

La topologie pair-à-pair ou *mesh* a aussi un coordinateur PAN. Cependant, elle diffère de la topologie en étoile dans le sens où tout nœud est capable de communiquer avec les autres aussi longtemps qu'ils sont dans la portée les uns des autres. La topologie p2p permet des réseaux plus complexes en terme de topologie, comme la topologie mesh. Un réseau p2p implémente un routage multi-sauts où chaque nœud route les messages de ses voisins vers le nœud destinataire. Le routage et la formation du réseau ne sont pas définis par la norme mais les couches supérieures basées sur IEEE 802.15.4.

Pour résumer, un PAN IEEE 802.15.4 se caractérise par un identifiant unique sur 16 bits dit adresse étendue. Cet identifiant permet la communication entre les équipements dans le réseau en utilisant les adresses courtes et autorise les transmissions entre les équipements de divers réseaux indépendants. C'est le coordinateur qui, à la formation du réseau choisit l'identifiant et aussi le canal sur lequel les nœuds doivent émettre. Le mécanisme par lequel les identifiants sont choisis ne fait pas partie de la norme [IEEE Computer Society, 2005].

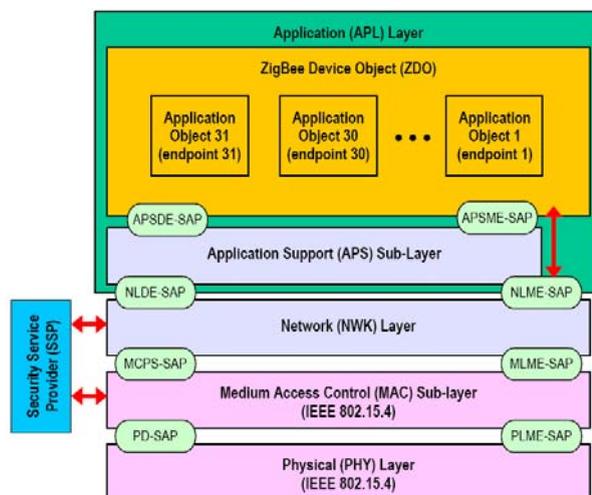
### 1.5.4 Zigbee

Promu par la Zigbee Alliance, Zigbee a pour but de compléter le IEEE 802.15.4 en spécifiant les couches suivantes : une couche réseau et une couche application. Ces diverses couches sont illustrées par la figure 1.7.

#### La couche réseau

La couche réseau est responsable de [Kinney *et al.*, 2003] :

- la création d'un réseau fonctionnel ;
- la jonction et la sortie des nœuds du réseau ;
- la configuration d'un nœud avec les paramètres du réseau ;
- l'adressage des nœuds rejoignant le réseau ;
- synchronisation des nœuds ;
- sécurisation des communications ;
- Le routage des trames vers leurs destinataires qui consiste en un routage hiérarchique. Pour plus de flexibilité, la spécification permet aux développeurs de définir des algorithmes adaptés à leurs applications mais aussi d'autres méthodes plus répandues comme *Ad hoc On-Distance Demand Vector* AODV.

FIGURE 1.7 – Spécification de la pile de protocoles Zigbee. Source [Kinney *et al.*, 2003]

La couche réseau du Zigbee supporte en natif les topologies : étoile, arborescente.

### La couche Application

Cette couche comporte plusieurs sous-couches qui sont : *Application objets*, *Zigbee Device objects (ZDO)* et *Application support sub-layer*.

**Application objets** permet à des fabricants de matériels Zigbee de définir des profils spécifiques à des applications.

**Zigbee Device Objects** inclut :

- La définition du rôle d'un appareil dans un réseau Zigbee. Ces rôles sont :
  1. le coordinateur Zigbee : forme la racine de l'arborescence du réseau et peut jouer le rôle de pont entre d'autres réseaux comme les réseaux WiFi, Bluetooth. Il y a exactement un seul coordinateur dans chaque réseau. Il stocke les informations sur le réseau ;
  2. un routeur Zigbee : route les données vers les autres nœuds suivant un algorithme donné. Il ne peut pas rentrer en mode veille ;
  3. le Zigbee End Device : a juste assez de fonctionnalités pour discuter avec le nœud parent (un coordinateur ou un routeur). Cette relation permet au nœud de dormir un temps significatif en conservant la durée de vie de la batterie. Il requiert moins de ressources qu'un routeur ou un coordinateur.
- l'initiation et ou la réponse à des requêtes d'association ;
- l'établissement de communications sécurisées entre les nœuds du réseau à l'aide d'une des méthodes de chiffrement à clé public ou privé.

**Application support sublayer** est chargé de la maintenance des tables de liaison qui permettent d'associer deux appareils en fonctions des services fournis et de leurs besoins et aussi de faire suivre les messages entre ces deux appareils. Cette sous-couche permet aussi

## 1.5. LA COMMUNICATION DANS LES RÉSEAUX DE CAPTEURS

Bande	Portée	Débit	Remarques
120-150 KHz (LF)	10 cm	Faible	Identification des animaux
13.56 MHz (HF)	1 m	De faible à modérée	Smart Card (MIFARE, ISO/IEC 14443)
433 MHz (UHF)	1-100 m	Modéré	Application de défense avec des tags actifs
865-868 MHz (Europe) 902-928 MHz (Amérique du Nord)	1-2 m 1-2 m	De modéré à élevé	Divers standard
2450-5800 MHz	1 - 2 m	Élevé	Même fréquence que le WiFi, Bluetooth
3.1 - 10 GHz	jusqu'à 200 m	Élevé	Requiert des tags semi-actifs et actifs

TABLE 1.8 – Caractéristiques de la transmission RFID. Source[Schwarz, 2013]

la découverte qui consiste à déterminer les autres appareils opérant dans la portée d'un nœud donné et quels services ils offrent [Kinney *et al.*, 2003].

### 1.5.4.1 Localisation par Zigbee

Les applications basées sur le Zigbee sont très nombreuses et cette pile vise la domotique, les automatisations dans les entreprises, etc.

Pour la localisation, Zigbee est exploité de la même façon que les normes précédentes. Il peut servir de support pour des méthodes [Corral *et al.*, 2008, Santinelli *et al.*, 2009] de triangulation, de proximité [Baronti *et al.*, 2007] et de fingerprinting [Martin *et al.*, 2013].

### 1.5.5 RFID/NFC

La radio-identification (RFID) est une technologie à courte portée destinée principalement à l'identification d'objets munis d'un marqueur RFID. Ces marqueurs sont lus par la suite par un lecteur RFID. De ce fait, il existe deux types de tags RFID actif et passif dont les bandes de fréquence avec les débits correspondants sont résumés dans le tableau 1.8.

La communication en champ proche est une technologie de transmission à courte portée (approximative de 10 cm) conçue pour permettre l'échange de plusieurs types d'informations entre deux appareils compatibles NFC, ou entre un appareil compatible NFC et un marqueur RFID actif ou passif (ISO/IEC 14443 A,B) qui sont tenus l'un à côté de l'autre [Schwarz, 2013]. Un appareil compatible NFC peut fonctionner selon les trois modes suivants :

1. émulation de carte ou mode passif : l'appareil se comporte comme une carte passive sans contact conforme aux standards RFID. Quatre types de tags ont été définis et résumés dans le tableau 1.9.

## 1.5. LA COMMUNICATION DANS LES RÉSEAUX DE CAPTEURS

	Type 1	Type 2	Type 3	Type 4
ISO/IEC	14443 A	14443 B	JIS 6319-4	14443 A/B
Débit	106 Kbits/s	106 Kbits	212, 424 Kbits/s	106/212/424 Kbits/s
Mémoire	96 bits, extensible à 2 KB	48 Bits, extensible à 2 Kbyte	Variable, max 1 MB	Variable, max 32 KB
Anti-collision	Non	Oui	Oui	Oui

TABLE 1.9 – Les types de tags NFC. Source [Schwarz, 2013].

Norme	Polling / Listening	Codage	Modulation	Débit	Fréquence porteuse
NFC-A	Polling Listening	Miller Modifié Manchester	ASK 100% Load Modulation (ASK)	106 Kbit/s 106 Kbit/s	13.56 MHz 13.56 MHz $\pm$ 848 Khz
NFC-B	Polling Listening	NRZ-L NRZ-L	ASK 10% Load Modulation (BPSK)	106 Kbits/s 106 Kbits/s	13.56 MHz 13.56 MHz $\pm$ 848 KHz
NFC-F	Polling Listening	Manchester Manchester	ASK 10% Load Modulation (ASK)	212/424 Kbits/s 212/424 Kbits/s	13.56 MHz 13.56 MHz (sans sous- porteuse)

TABLE 1.10 – Les modes de communication défini pour un appareil NFC. Source [Schwarz, 2013].

2. pair-à-pair : ce mode permet l'échange de données entre deux appareils NFC. L'initiateur (polling device) requiert peu de puissance comparé au mode lecture/écriture car la cible (listener) utilise sa propre source de puissance ;
3. lecture/écriture ou mode actif : l'appareil NFC joue le rôle actif et lit et écrit une puce passive compatible RFID.

NFC opère dans la bande de fréquence 13.56 MHz  $\pm$  848 Khz en utilisant trois modes de communications : NFC-A, NFC-B et NFC-F. Un appareil désirant communiquer commence par scanner les appareils compatibles NFC en émettant une requête de connexion respectivement selon les mode NFC-A (compatible ISO/IEC 14443 A) , NFC-B (compatible ISO/IEC 14443 B) et NFC-F (compatible JIS X 6319-4). La connexion est établie suivant la réponse obtenue. Une description de ces modes de communication est donnée dans la Table 1.10.

### Localisation par NFC/RFID

L'avantage du positionnement par NFC/RFID réside dans la courte portée de ces technologies de transmission. En effet, un déploiement soigneusement effectué des tags permet d'avoir un positionnement par proximité de niveau centimétrique. Comme indiqué pour les autres technologies de transmission, la localisation peut être effectuée aussi bien par

méthode géométrique que par fingerprinting [Mautz, 2012].

## 1.6 Modélisation des réseaux de capteurs

Un<sup>2</sup> réseau de capteurs peut être modélisé par un ensemble,  $\mathfrak{R}$ , de  $n$  nœuds ( $n_i$ ) :

$$\mathfrak{R} = \{n_i; i = 0, \dots, N - 1\} \quad (1.1)$$

À un instant donné, un sous-ensemble de nœuds,  $n_k$  avec  $k \in 0..m, m \leq N$ , sont à des positions fixes,  $x_k \in \Omega$ . Et un autre sous-ensemble de nœuds mobiles,  $n_j$  avec  $j \in 0..N - m$  se déplaçant suivant une trajectoire  $x_j(t)$ .  $\Omega$  représente la zone de déploiement du réseau, et est un sous-ensemble de  $R^n$ ,  $n$  étant de dimension, souvent égale à 2 et parfois 3. Plusieurs scénarios peuvent être envisagés pour la mobilité des nœuds [Aschenbruck *et al.*, 2010]. Par exemple le nœud se déplace suivant une modélisation des déplacements en cas de catastrophes, une marche en situation normale, ou un déplacement aléatoire, etc.

BonnMotion [BonnMotion Team, 2013] est un programme permettant de générer des données, pour les trajectoires des nœuds pour plusieurs simulateurs : NS-2, NS-3, GlomoSim/QualNet, COOJA, MixiM et ONE [Aschenbruck *et al.*, 2010].

Du point de vue du positionnement, nous distinguons les sous-ensembles :

- $P = \{nr_i, i = 0..p\}$ , l'ensemble des  $p$  nœuds de référence, situés à des positions connues. Ces positions ne sont pas susceptibles de changer sauf indisponibilité du nœud de référence ou redéploiement exigé par l'application, ou aussi si on considère que le nœud est mobile, à l'image des *pseudolites* dans les systèmes pseudo-satellitaires. Dans ce cas, comme pour les satellites, la position est connue à chaque instant.
- $Q = \{ni_i, i = 0..q\}$ , l'ensemble des  $q$  nœuds cherchant à estimer leur position.
- Avec  $P \cup Q = \mathfrak{R}$ . Il n'est pas exclu qu'une fois qu'un nœud ait estimé sa position, qu'il devienne un nœud de référence. Cependant, du fait de la propagation des erreurs d'estimation, un poids est assigné à la position. Le poids indique si l'erreur est suffisamment petite pour que le nœud puisse servir de base aux autres pour déterminer leur position.

Plusieurs hypothèses peuvent être faites par les méthodes de positionnement sur la mobilité des nœuds et le nombre de nœuds de référence disponibles dans le réseau à l'initialisation de la méthode. Plusieurs paramètres peuvent être déterminés par simulation à savoir l'influence du nombre  $p$  de nœuds de références sur la qualité du positionnement pour un algorithme de positionnement utilisant une ou des normes de transmissions que nous venons de décrire dans un environnement donné. Il serait aussi important de déterminer le nombre de nœuds à position inconnue que le système peut positionner sans dégrader les performances attendues et annoncées par un tiers. Et si ce nombre est dépassé, quel est le nombre de nœuds de référence à déployer pour atteindre les exigences ? Combien de

---

2. Cette section s'inspire largement du livre de E. Fleury et D. Simplot-Ryl [Fleury et Simplot-Ryl, 2009]. Les passages tirés textuellement sont mis entre guillemets et les reformulations sont suivies par la référence au livre. Lorsque d'autres travaux sont aussi cités nous mentionnons explicitement les références.

nœuds mobiles transformer en nœud de référence ? Et comment les déployer pour ne pas perturber les autres ?

« La liaison radio entre deux nœuds,  $n_i$  et  $n_j$  est caractérisée par une fonction  $l(x_i, x_j)$ , notée  $l_{ij}$ . Cette liaison peut varier dans le temps dans les cas suivants : i) les nœuds sont mobiles (on a alors  $x_i(t), x_j(t)$ ); ii) le canal est non stationnaire; iii) ou par combinaison des deux phénomènes. Il s'écrit alors  $l_{ij}(t)$  ».

Le réseau  $\mathfrak{R}$  peut être représenté par un graphe, dont les sommets sont les nœuds, et les arêtes les liaisons valides :  $G = (\mathfrak{R}, L)$ , où :

$$L = \{(x_i, x_j) \in \mathfrak{R} \mid l_{ij} \neq 0\} \quad (1.2)$$

Ainsi, les propriétés des réseaux de capteurs et des problèmes, dans notre cas la localisation dans les environnements à fortes perturbations des liens radio, peuvent être étudiées en se servant des propriétés du graphe associé au réseau et des outils théoriques ou algorithmiques classiques utilisés en théorie des graphes.

Nous allons présenter le modèle du graphe unitaire, largement utilisé dans la simulation, puis la prise en compte de phénomènes plus complexes intervenant dans le milieu de propagation et à prendre en compte dans une simulation réaliste.

## 1.6.1 Modèle du Graphe Unitaire (*Unit Disk Graph*)

### 1.6.1.1 Définition

Le modèle du graphe unitaire, ou modèle géométrique, est un modèle simple défini autour d'une distance supposée  $R$  appelée la portée radio. Ce modèle considère pour tout nœud  $n_i$  que certains nœuds sont à portée et d'autres hors de portée. Il est basé sur des hypothèses fortes qui lui confèrent ses bonnes propriétés [Fleury et Simplot-Ryl, 2009]. Le graphe ( $G$ ) est dit géométrique car l'existence d'un lien radio dépend uniquement de la distance géométrique entre l'émetteur et le récepteur. Ainsi la zone de communication de chaque nœud est un disque centré de rayon  $R$ .

La liaison  $l_{ij}$  est alors définie par :

$$l_{ij} : \quad \Omega^2 \mapsto B = \{0, 1\}, \quad (1.3)$$

$$(x_i, x_j) \mapsto l(x_i, x_j) = \begin{cases} 1 & \text{si } \|x_i - x_j\| \leq R, \\ 0 & \text{sinon} \end{cases} \quad (1.4)$$

où  $\|x_i - x_j\|$  est la distance euclidienne ( $d_{ij}$ ) dans  $R^n$  entre les positions  $x_i$  et  $x_j$ .

Ce modèle, bien que faisant des hypothèses fortes, est utilisé pour la modélisation et la simulation de protocoles.

## 1.6.2 Propriétés

Les hypothèses ont été rassemblées sous la forme des six axiomes suivants :

$$\text{stationnarité temporelle : } l_{ij}(t) = l_{ij} \quad (1.5)$$

$$\text{indépendance : } l_{ij} = f(x_i, x_j) \quad (1.6)$$

$$\text{liaison commutée (on/off) : } l_{ij} \in \{0, 1\} \quad (1.7)$$

$$\text{symétrie : } l_{ij} = l_{ji} \quad (1.8)$$

$$\text{isotropie : } l_{ij} = f(x_i, d_{ij}) \quad (1.9)$$

$$\text{homogénéité : } l_{ij} = f(d_{ij}) \quad (1.10)$$

La stationnarité temporelle traduit le fait que la liaison ne varie que lorsque le nœud est mobile. Elle ne prend, cependant pas en compte les perturbations de la liaison induites par les obstacles fixes ou mobiles.

L'indépendance traduit le fait qu'une liaison  $l_{ij}$  ne dépend que du bruit sur le lien radio entre les nœuds  $i$  et  $j$ . Le bruit est mesuré par le rapport signal bruit (RSB, aussi noté SNR, pour *signal to noise ratio*). Cette hypothèse ne prend pas en compte la contribution des nœuds voisins en terme des interférences, les brouillages, la corrélation des effets de masques, etc.

Soit  $s\bar{n}r_{ij}$ , le SNR moyen associé à la liaison  $l_{ij}$ .

$$s\bar{n}r_{ij} = \bar{h}_{ij} \cdot \frac{P_i}{N_j} \quad (1.11)$$

où  $P_i$  et  $N_j$  sont respectivement les puissances d'émission et de bruit de réception associées à la liaison  $l_{ij}$ . La fonction  $\bar{h}_{ij} \in [0, 1]$ , définie sur  $\Omega^2$  caractérise l'atténuation du canal de propagation entre toute paire de points  $(x_i, x_j)$  :

$$\bar{h} : \quad \Omega^2 \longmapsto [0, 1] \quad (1.12)$$

$$(x_i, x_j) \rightarrow \bar{h}(x_i, x_j) \quad (1.13)$$

La fonction  $\bar{h}(x_i, x_j)$  est choisie pour être continue, strictement décroissante avec  $d_{ij}$ . Cependant dans un environnement réel, des discontinuités fortes peuvent être observées à causes des obstacles (mûrs en béton par exemple) et la décroissance stricte n'est pas garantie à cause des phénomènes de chemins multiples générés par la réflexion des ondes sur ces mêmes obstacles [Fleury et Simplot-Ryl, 2009].

L'axiome de liaison commutée traduit le fait que l'existence d'un lien entre deux nœuds dans la même portée dépend d'un SNR seuil  $s\bar{n}r_{lim}$ . Ce SNR est déterminé à partir du théorème de Shannon :

$$C_{ij} = W \cdot \log_2(1 + s\bar{n}r_{ij}) \quad (1.14)$$

Où  $W$  est la bande passante du système (en Hz).

Le SNR théorique minimum, pour un système fonctionnant au débit binaire  $D$  est donné par :  $s\bar{n}r_{lim} = 2^{D/W} - 1$ . Les liaisons sont alors définies par :

$$l_{ij} : \quad \Omega^2 \longmapsto B = \{0, 1\}, \quad (1.15)$$

$$(x_i, x_j) \longmapsto l(x_i, x_j) = \begin{cases} 1 & \text{si } s\bar{n}r_{ij} \geq s\bar{n}r_{lim} \\ 0 & \text{sinon} \end{cases} \quad (1.16)$$

Le LQI (Link Quality Indicator) est une autre fonction permettant d'indiquer la qualité du lien radio.

La symétrie, basée sur l'indépendance, stipule les mêmes phénomènes entre les deux nœuds et se base sur la réciprocité du SNR :

$$sn\bar{r}_{ij} = sn\bar{r}_{ji} \quad (1.17)$$

L'isotropie est basée sur l'existence d'une relation directe entre  $d_{ij}$  et  $\bar{\gamma}_{ij}$  pour un nœud donné  $n_i$ . De cet axiome découle la notion de portée  $R$  et de voisinage circulaire.

L'homogénéité garantit la même portée à tout nœud du réseau.

### 1.6.3 Bilan de la liaison

Le bilan de la liaison entre un émetteur et un récepteur permet, à partir de la puissance reçue à une distance  $d_{ij}$  par le récepteur, de déterminer l'impact du milieu de propagation (affaiblissement moyen, effets de masques, évanouissement non sélectifs) et des paramètres des nœuds, notamment la sensibilité et la configuration des systèmes antennaires (diagramme de rayonnement, gain/directivité, polarisation) utilisées par les deux nœuds, dans la transmission de données sur la liaison. Plusieurs hypothèses peuvent être faites sur les systèmes antennaires, dont une symétrie entre les nœuds et les phénomènes de propagation [Fleury et Simplot-Ryl, 2009].

La formule de Friis [Rappaport, 2002] permet d'établir un premier bilan de la façon suivante :

$$P_j = \left( \frac{\lambda}{4\pi d_{ij}} \right)^\eta \cdot G_i(\theta_{ij}, \phi_{ij}) \cdot G_j(\theta_{ji}, \phi_{ji}) P_i \quad (1.18)$$

$\lambda$  est la longueur d'onde du signal (inversement proportionnelle à la fréquence) facilement déterminable en fonction de la technologie de transmission utilisée.

$\eta$  est l'exposant de perte dépendant de l'environnement. C'est un des paramètres importants du milieu. Il n'est pas facile à déterminer et nécessite une calibration qui est effectuée par des campagnes de mesures.

$G_i(\theta_{ij}, \phi_{ij})$  et  $G_j(\theta_{ji}, \phi_{ji})$  sont respectivement les gains des antennes de l'émetteur et du récepteur. Les angles  $\theta_{ij}, \phi_{ij}$  indiquent la direction  $\vec{n}_i n_j$  par rapport à l'axe principal de l'antenne  $i$  et réciproquement pour les angles  $\theta_{ji}, \phi_{ji}$ .

« Le gain d'une antenne permet de mesurer le rendement de cette antenne dans une direction donnée. En effet, la puissance  $p_f$  fournie à l'antenne par le nœud est rayonnée en puissance effective  $p_e$ . De ce fait, le rendement associé à l'antenne est  $r = p_e/p_f$ .

Cependant, les antennes ne rayonnent pas toutes de la même façon dans toutes les directions suivant leur diagramme de rayonnement. On distingue l'antenne :

- isotrope : est une antenne théorique qui rayonne de la même façon dans toutes les directions de l'espace. Ainsi, la densité de puissance est la même quelque soit l'angle solide considéré :  $P_e/4\pi$ . Le diagramme isotrope sert de référence pour quantifier le rayonnement des autres types d'antennes. Son gain est de 1 ou de 0 dBi (décibel par rapport à l'antenne isotrope) ;

- omnidirectionnelle : a un rayonnement uniforme dans un seul plan. Par exemple, le rayonnement est uniforme dans le plan horizontal, avec une densité de puissance différente.
- directive : privilégie une direction donnée liée à une ouverture (angle total où la densité de puissance est entre la valeur maximale et la moitié de cette valeur). Il s'agit alors de rayonner la majeure partie de la puissance dans une ouverture angulaire donnée suivant l'hypothèse que cette puissance est fixe durant le temps d'émission.
- agile : est une antenne pouvant changer son diagramme de rayonnement en fonction des conditions de fonctionnement. C'est le cas des antennes multi-faisceaux ou des antennes dites intelligentes pouvant changer dynamiquement de diagramme.

Ainsi, le gain d'une antenne est déterminé par :

$$G(\theta, \phi) = D(\theta, \phi) \times P_e/P_f = \frac{\Delta P(\theta, \phi)}{P_f/4\pi} \quad (1.19)$$

$D(\theta, \phi)$  mesure la directivité de l'antenne. À savoir la densité de puissance obtenue dans une direction particulière de l'espace par rapport à la densité créée par une antenne isotrope. Tout en sachant que quelque soit le diagramme de l'antenne, la puissance totale rayonnée est  $P_e$ . La directivité est donnée par :

$$\Delta P = \frac{\Delta P(\theta, \phi)}{P_e/4\pi} \quad (1.20)$$

En résumé, pour un même diagramme de rayonnement, le gain peut être différent en fonction des directions. Par exemple une antenne omnidirectionnelle peut en théorie avoir un gain maximum entre 1 (cas isotrope) et infini (cas d'une ouverture verticale nulle). Cela étant dit, on pourra garder comme postulat que le gain maximum d'une antenne est globalement lié à sa taille, donc dans notre cas d'étude on restera sur des valeurs faibles.

Une autre notion importante est la polarisation des antennes. En effet le champ magnétique rayonné possède une forme vectorielle. Le vecteur de champ (électrique ou magnétique de façon équivalente) se propage alors en absence d'obstacles en libre droite, mais l'orientation de ce vecteur peut varier au cours du déplacement. C'est la structure de l'antenne d'émission qui va dicter cette orientation : une polarisation rectiligne correspond à un vecteur gardant constamment le même axe principal, alors qu'au contraire une polarisation circulaire rend compte d'un champ dont le vecteur tourne en permanence. Les antennes étant réciproques, en réception une antenne à polarisation rectiligne ne captera que la projection du vecteur sur son axe. On parle alors de rendement de polarisation, pouvant aller de l'optimum (égal à 1) pour deux antennes à polarisation rectiligne avec le même alignement, à un rendement nul pour deux antennes dont les polarisations sont perpendiculaires. On notera que la combinaison circulaire-circulaire permettrait de garantir un rendement maximal sans soucis d'alignement, mais la mise en œuvre de telles antennes est beaucoup plus complexe. »

$P_i$  la puissance du signal d'émission. Ainsi, on peut déduire la puissance du signal reçu  $P_j$ .

De la formule de Friis découle une forme simple de l'affaiblissement moyen du lien radio  $a_{ij}$ . Nous allons voir un peu plus loin que la modélisation de cet affaiblissement peut être affinée.

#### 1.6.4 Modélisation déterministe réaliste

La modélisation de la fonction d'atténuation  $\bar{h}_{ij}$  peut s'envisager sous deux approches complémentaires : déterministe et stochastique. Pour cela,  $\bar{h}_{ij}$  est usuellement décomposée en trois composantes : l'affaiblissement moyen noté  $a_{ij}$ , les effets de masques (*shadowing*), les évanouissements (*fading*).

#### 1.6.5 Affaiblissement moyen

L'affaiblissement moyen,  $a_{ij}$  est donné par :

$$a_{ij} = \frac{P_i}{P_j} = \left( \frac{4\pi}{\lambda} \cdot d_{ij} \right)^{-\eta} \cdot G_i^{-1}(\theta_{ij}, \phi_{ij}) \cdot G_j^{-1}(\theta_{ji}, \phi_{ji}). \quad (1.21)$$

Où  $\lambda$  désigne la longueur d'onde du signal,  $G_i(\theta_{ij}, \phi_{ij})$ ,  $G_j(\theta_{ji}, \phi_{ji})$  respectivement les gains des antennes de l'émetteur et du récepteur. À partir de cette composante déterministe, la distance  $d_{ij}$  peut être estimée.  $\eta = 2$  modélise la propagation en espace libre qui dispose de conditions idéales.

Par souci de simplicité, dans les simulations, les gains sont maximum à savoir 1. Ainsi l'affaiblissement devient :

$$a_{ij} = \left( \frac{4\pi}{\lambda} \cdot d_{ij} \right)^{-\eta} \quad (1.22)$$

Une interpolation de  $a_{ij}$  est donnée par [Fleury et Simplot-Ryl, 2009] :

$$a_{ij} = a_0 \cdot \left( \frac{d_{ij}}{d_0} \right)^{-\eta}, \quad (1.23)$$

où  $a_0$  et  $\eta$  sont deux constantes, caractéristiques de l'environnement. Le coefficient  $\eta$  est en général supérieur à 2 et inférieur à 6. Mais peut être inférieur à 2 dans les cas où des obstacles guident l'onde, comme les couloirs en béton.

#### 1.6.6 Modélisation stochastique

##### 1.6.6.1 Les effets de masque (Shadowing)

« La composante stochastique est appelée *effet de masque* et complète le modèle empirique par des variations locales de puissance reçue, dues à la présence (ou l'absence) d'obstacles. L'atténuation du canal  $\bar{h}_{ij}$  n'est donc plus uniquement liée à l'affaiblissement moyen  $a_{ij}$ , mais l'expérience appuyée par le théorème central limite [Neskoic *et al.*, 2000]

justifie l'utilisation d'un bruit multiplicatif ayant une densité de probabilité (d.d.p.) log-normale  $\bar{s}_{ij}$ , telle que :

$$\bar{h}_{ij}^{(dB)} = a_{ij}^{(dB)} + \bar{s}_{ij}^{(dB)}, \quad (1.24)$$

Il est important de noter que l'effet de masque étant lié aux obstacles immobiles, chaque lien  $l_{ij}$  est invariant pour une réalisation donnée et sans mobilité des nœuds. Cette modélisation induit alors une déformation géométrique du modèle à disque. Certaines liaisons de longueur  $d_{ij} > R$  sont actives, alors que d'autres de longueur  $d_{ij} < R$  restent inactives. »

Ce modèle appelé log-normal est très largement utilisé dans la littérature, tant en simulation qu'en modélisation. La formulation la plus courante est donnée par l'équation (1.25)

$$P_r(d) = P_0(d_0) - \eta 10 \log\left(\frac{d}{d_0}\right) + X_\sigma \quad (1.25)$$

Où  $P_r(d)$  est la puissance reçue à la distance  $d$  et  $P_0(d_0)$  la puissance reçue à une certaine distance de référence  $d_0$ . Pour simplifier la simulation, les valeurs moyennes ou les distributions de probabilités peuvent être définies pour la puissance.

Il faut noter que dans la réalité, la mesure de la puissance du signal reçu est très dépendant du constructeur du matériel. Le RSSI et le LQI sont les deux notions fréquemment utilisées dans la littérature pour « mesurer » la puissance.

Le RSSI est une indication de la force du signal. Il ne mesure pas « la qualité » ou la « correctitude » du signal.

Le *Link Quality Indicator* est une métrique mesurant la qualité du signal reçu. Il indique le degré de facilité de la démodulation du signal reçu. En déterminant l'erreur totale entre une valeur idéale et le signal reçu sur les 64 symboles suivant immédiatement le mot de synchronisation. Ainsi, c'est un indicateur de la qualité du lien dont une valeur élevée indique une mauvaise liaison et une valeur faible une bonne liaison. Par exemple, pour une modulation FSK ou GFSK, le récepteur va mesurer la fréquence de chaque « bit » et la compare avec la fréquence attendue en se basant sur la fréquence du canal, de la déviation et de l'offset de la fréquence mesuré. Lorsque d'autres modulations sont utilisées, par exemple PSK ou ASK ce sont respectivement les phases et les angles qui sont mesurées. Le LQI ne tiens pas compte de la puissance du signal, mais la qualité du signal est liée à la force du signal. Ceci par ce que qu'un signal fort a plus de chance d'être moins affecté par le bruit et de ce fait sera vu comme « clair » ou plus correct par le récepteur [TI, 2013].

Par exemple, pour les réseaux IEEE 802.15.4, la valeurs du LQI pour le signal le plus fort et le plus faible sont respectivement de  $0x00$  à  $0xFF$  [IEEE Computer Society, 2005].

« La sensibilité caractérise le niveau minimal de puissance requis en entrée du récepteur (donc l'antenne) pour assurer un fonctionnement correct du système :

$$S(dBW) = 10 \cdot \log(\bar{\gamma}_{min}) + 10 \log(k.T.o.B) + F(dB) \quad (1.26)$$

Cette expression simple de la sensibilité donnée en échelle logarithmique (dBW = décibels rapportés à 1 watt) est la somme de trois termes principaux que nous allons expliciter.

Le premier terme,  $10 \cdot \log(\bar{\gamma}_{min})$ , est le rapport signal à bruit (SNR) exprimé en décibels. Comme expliqué en début de chapitre, cette grandeur représente le niveau de puissance

Environnement	$\sigma(dB)$
Outdoor	4 à 12
Bureau, Hard Partition	7
Bureau, Soft Partition	9.6

TABLE 1.11 – Exemple de valeurs typiques pour  $\sigma$ .

par rapport au niveau de bruit. Pour fixer la sensibilité d'un récepteur, on a donc besoin de connaître le SNR nécessaire pour garantir un bon fonctionnement du système, qui est déterminé par le seuil de taux d'erreur acceptable sur une communication.

Le second terme,  $10 \cdot \log(k \cdot T_o \cdot B)$ , donne le niveau de puissance (en dBW) du bruit interne du récepteur. Ce bruit dit thermique, est un bruit blanc additif gaussien (noté AWGN, pour *additive white gaussian Noise*). Il est présent à l'entrée du récepteur et c'est lui qui limite intrinsèquement la capacité d'une liaison radio.

Le dernier terme,  $F(dB)$ , représente le facteur (ou figure) de bruit du récepteur. Dans toute architecture de récepteur radio, les composants permettant de passer du signal capté par l'antenne à une information numérique interprétable engendrent une augmentation du bruit. En particulier, les amplificateurs nécessaires pour relever l'amplitude du signal reçu avant numérisation, ne sont pas parfaits et ajoutent du bruit par rapport au bruit thermique qui constitue la limite basse du niveau de bruit d'un récepteur. Il en découle donc un rapport entre le SNR seuil,  $\bar{\gamma}_{lim}$  au niveau de l'antenne et  $\bar{\gamma}_{min}$ , au travers du facteur de bruit du récepteur. Dès lors, si l'on veut intégrer les contraintes sur les caractéristiques du récepteur, une première approche consiste à prendre en compte cette correction du SNR seuil, autrement dit à considérer que le niveau minimal de puissance reçue  $P_j$  garantissant la liaison est égal à la sensibilité du récepteur. Par la suite nous ne conserverons pas le détail de cette prise en compte, mais ce facteur de bruit peut être intégré pour plus de réalisme soit au niveau du calcul du niveau de bruit au récepteur,  $N_j$ , soit dans l'expression de la capacité du canal radio. Cependant cela suppose de connaître le type d'interface radiofréquence utilisée, ce qui n'est pas toujours possible dans une étude générale des réseaux sans fil. »

$\eta$  est l'exposant de perte. Des valeurs typiques [Fall et Varadhan, 2002] pour  $\eta$  en environnement :

- *Outdoor* :
  - Espace ouvert : 2
  - Zone urbaine avec effets de masque : 2.7 à 5
- *Indoor* :
  - Ligne directe (*Line of Sight*) : 1.6 à 1.8
  - Obstrué : 4 à 6

$X_\sigma$  est une variable aléatoire log-normale de variance  $\sigma^2$  qui représente les effets de masque. Le tableau (1.11) donne quelques valeurs typiques de  $\sigma$  en dB.

L'ITU [Sarkar *et al.*, 2003] a défini un autre modèle pour propagation à l'intérieur donnée par la formule :

$$P = 20 \log f + \eta \log(d) + c(k, f) - 28 \quad (1.27)$$

Où  $f$  est la fréquence du signal,  $c$  est une valeur empirique représentant le *floor loss*

*penetration actor* et  $k$  est le nombre d'étages séparant un émetteur d'un récepteur. La valeur du coefficient de perte de chemin  $\eta$  est déterminé de façon empirique et se trouve généralement entre 20 et 30. Ce type de modèle est dénommé par WAF (*Wall Attenuation Factor*) ou FAF (*Floor Attenuation Factor*).

### 1.6.7 Les évanouissements non sélectifs

« Le processus de masque a permis d'intégrer les variations spatiales de la fonction d'atténuation de canal, mais n'intègre pas les variations rapides de signal qui résultent de la sommation incohérente des échos du même signal source, dus aux multitrajets. Le phénomène d'évanouissement qui en résulte est bien connu. Le moindre mouvement de l'émetteur, du récepteur, ou même simplement d'un obstacle, peut modifier le schéma de sommation et introduire des variations rapides dans la puissance du signal radio reçu. La modélisation déterministe des évanouissements est impossible car ils dépendent de phénomènes *microscopiques*. Ils sont donc en général caractérisés par un processus aléatoire, que l'on notera  $e_{ij}(t)$  et qui vient compléter les deux termes précédents pour donner une atténuation instantanée :

$$\bar{h}_{ij}^{(dB)} = a_{ij}^{(dB)} + \bar{s}_{ij}^{(dB)} + e_{ij}^{(dB)}(t) \quad (1.28)$$

On note ici que  $a_{ij}^{(dB)}$  et  $\bar{s}_{ij}^{(dB)}$  ne dépendent pas du temps.

Le terme  $e_{ij}(t)$  peut être considéré comme un processus non corrélé spatialement car les évanouissements varient rapidement et la longueur de corrélation est petite devant la distance moyenne entre les nœuds.

Les statistiques à l'ordre 1 du processus d'évanouissement  $e_{ij}(t)$  sont bien connues. Lorsqu'aucun chemin de transmission n'est prépondérant (canal dit de Rayleigh), la densité de probabilité de puissance suit une loi exponentielle :

$$f_e(e = \xi) = \exp(-\xi) \quad (1.29)$$

D'autres modèles plus complets existent, comme les canaux de Rice ou de Nagakami-m (voir par exemple [Simon et Alouini, 2004] pour une description des différents modèles de canaux). »

## 1.7 Simulation des réseaux de capteurs

La simulation des phénomènes de propagation dans un environnement donné peut se faire suivant deux approches : la technique dite du tracé ou lancé de rayons et la méthode des éléments finis. À partir de cette simulation, la valeur de la puissance du signal ou les paramètres du modèle de propagation dans cet environnement peuvent être déterminés. Des paramètres de la méthode de positionnement peuvent aussi être déterminés.

La technique de tracé ou lancé de rayon consiste en une description fine de l'environnement en termes de matériaux, des structures et des perturbations du signal induites par ces

matériaux. Tous ces matériaux doivent être connus afin de mieux caractériser l'environnement. Les questions importantes à se poser sont : combien de phénomènes et de matériaux faudrait-il prendre en compte ? Plusieurs scénarios sont possibles dont l'usage d'une table de tous les matériaux et leurs caractéristiques. Un équilibre doit être trouvé entre nombre de phénomènes à prendre en compte, les matériaux et le temps de calcul. Tout ceci est pesé contre les attentes du domaine d'application en terme de précision de mise en place du modèle de propagation [Evennou, 2007].

Un modèle utilisé est donné par l'équation (1.30) en  $dBm$  :

$$p(d) = p(d_0) - 10\eta \log\left(\frac{d}{d_0}\right) + \sum_{i=0}^M n_i \times R_i + \sum_{i=0}^N n_i \times T_i \quad (1.30)$$

Où :

- $R_i$  représente les réflexions introduites par le matériels  $i$ . Avec  $M$  le nombre de matériaux sur lesquels il y a eu au moins une réflexion ;
- $T_i$  : les transmissions sur les matériels  $i$ .  $N$  est le nombre de ces matériaux.
- $p(d)$  : la puissance mesurée à la distance  $d$  ;  $p(d_0)$  est la valeur de la puissance mesurée à la distance  $d_0$ .

Une phase de calibration, avec des mesures effectuées dans plusieurs points du bâtiment permet de déterminer les coefficients du modèle présenté ci-dessus de la façon suivante [Evennou, 2007] :

$$\left[ \eta^{opt}, R_i^{opt}, T_i^{opt} \right] = \arg \min_{\eta, R_i, T_i} \left[ \left( p(d) - p(d_0) - 10\eta \log\left(\frac{d}{d_0}\right) + \sum_{i=0}^M n_i \times R_i + \sum_{i=0}^N n_i \times T_i \right)^2 \right] \quad (1.31)$$

La méthode par différence finie (ou éléments finis) se base sur une discrétisation de l'environnement en cellules élémentaires de petite taille, bien inférieure à la longueur d'onde, où le milieu peut être considéré de caractéristique homogène. La démarche est la suivante, comme décrite dans [Oudin, 2008] :

- discrétisation du milieu continu en sous-domaines ;
- construction de l'approximation nodale par sous-domaine ;
- calcul des matrices élémentaires correspondant à la forme intégrale du problème ;
- assemblage des matrices élémentaires ;
- prise en compte des conditions aux limites ;
- résolution du système d'équations.

Ces méthodes sont très précises mais sont coûteuses en temps de calcul lorsque la taille de l'environnement augmente.

## 1.8 Conclusion

Dans ce chapitre, nous avons présenté les réseaux de capteurs en insistant sur les difficultés de la transmission en environnement clos pour mieux appréhender les difficultés de réalisation d'un système de localisation dans ces milieux. Ces difficultés liés à l'usage des ondes électromagnétiques sont surtout dues à la propagation de ces ondes dans un milieu

## 1.8. CONCLUSION

---

donné. Nous avons présenté une modélisation de ces perturbations sous la forme d'une fonction d'atténuation  $h$  composée de trois composantes : l'affaiblissement, l'effet de masque et les évanouissements non sélectifs. Le chapitre suivant va approfondir les méthodes de positionnement.

## 1.8. CONCLUSION

---

## Chapitre 2

# La localisation par réseaux de capteurs

### 2.1 Introduction

La localisation permet l'assignation d'une position absolue ou relative à une entité en mouvement ou fixe. Une position absolue fait référence à une position globale comme les positions dérivées d'un système géodésique [Moritz, 1980] tandis qu'une position relative est définie de façon locale et valable dans une zone donnée. Remarquons qu'en fonction des systèmes de localisation, une relation peut exister entre ces deux types de positions [Grewal *et al.*, 2007].

La position ainsi obtenue est la pierre angulaire pour plusieurs applications. Elle permet à ces applications de fournir des services dits basés sur la position (*location-based services* [Schiller et Voisard, 2004] : LBS). Par exemple la navigation dans les espaces publics couverts comme les aéroports et, en même temps, la description des places visitées, fournir une publicité adaptée, etc. La position est un des éléments fondamentaux de la notion, plus globale, de contexte qui définit aussi d'autres types de services dits *context-aware services* [Schilit *et al.*, 1994, Abeywickrama et Ramakrishnan, 2012, Satoh, 2012].

On peut distinguer : i) la localisation à l'extérieur ; ii) la localisation à l'intérieur prenant en compte les espaces entièrement couverts et aussi les environnements dans lesquels la localisation externe est difficile ; iii) le dernier cas est une hybridation entre localisation externe et interne, le système est dans ce cas capable de décider de la solution à utiliser. Par exemple, effectuer une transition entre positionnement par GPS et par signal IEEE 802.11<sup>1</sup> dès que la qualité du signal GPS est insuffisante avec tous les changements que cela implique en terme d'acquisition de données et des algorithmes à utiliser [Zirari *et al.*, 2010]. Google Maps [Google Inc, 2011] est un exemple d'application notoire pour le positionnement par GPS et qui est de plus en plus utilisé pour le positionnement à l'intérieur de bâtiments, notamment dans les aéroports et centres commerciaux de certaines grandes villes.

La localisation à l'extérieur a atteint de meilleurs résultats et est exploitée par plusieurs applications grâce aux avancées dans le domaine des systèmes satellitaires géostationnaires,

---

1. Le terme WiFi sera utilisé à la place du IEEE 802.11 dans la suite.

## 2.2. DÉFINITIONS

---

comme le GPS [Kaplan et Hegarty, 2006] et Galiléo [Benedicto *et al.*, 2000] entre autres, ou la localisation par réseaux cellulaires [Darnell et Douglas, 1991, Kangas *et al.*, 2011, Zhao, 2002].

A contrario, la localisation à l'intérieur de bâtiment, qui est l'objet de ce chapitre, souffre des conditions particulièrement difficiles des milieux dits *indoor* et peine à avoir une exploitation commerciale comparable à celle offerte par les systèmes satellitaires de positionnement ou *Global Navigation Satellite Systems* GNSS [Hofmann-Wellenhof *et al.*, 2007]. Ces difficultés sont matérialisées par les perturbations du signal électromagnétique, et ont été largement étudiées dans [Fleury et Simplot-Ryl, 2009], aussi il faudra adapter chaque système de positionnement à des conditions d'exploitation particulières.

L'amélioration de la vie des personnes handicapées et âgées dans leur milieu de vie, au travail, dans les moments de loisirs, fait partie des services basés sur la position. Ce domaine d'application est connu sous le nom *Ambient Assited Living* [Steg *et al.*, 2006, Kleinberger *et al.*, 2007, Sun *et al.*, 2009]. La première étape est de déterminer la position ou le contexte et la seconde étape consiste à fournir les services définis en collaboration avec les autorités sanitaires, les personnes pour lesquelles le système va être conçu. Ainsi, les défis liés à la mise en place de ces services sont nombreux.

Dans ce chapitre nous allons étudier les difficultés liées à la localisation des entités de façon générale.

## 2.2 Définitions

Le processus d'assignation d'une position à une entité en se basant sur des informations obtenues sur et/ou de son environnement est dénommée par plusieurs termes en fonction des technologies utilisées et des auteurs. Ces termes sont définis dans cette section. Bien entendu, les définitions données différeront d'un domaine à un autre. Cela est valable surtout pour localisation.

Une position peut être exprimée de façon physique ou symbolique. Néanmoins, les positions symboliques peuvent être utilisées pour « augmenter » les positions physiques, à l'image de la réalité augmentée.

Les positions physiques sont exprimées sous la forme de coordonnées qui identifient un point sur une carte et peuvent être globales ou locales. Une position symbolique est un élément du langage naturel servant à indiquer une position. Par exemple : « bibliothèque », « à côté de », « la porte de sortie » [Mautz, 2012].

La combinaison de positions physiques et symboliques permet de définir un modèle de position, définissant la sémantique de la position en fonction de l'application. Le modèle ainsi défini doit prendre en compte le fait que dans tout système de positionnement, abstraction faite du type de coordonnées utilisées, il devrait exister un ensemble de coordonnées qui représentent des zones inaccessibles physiquement pour l'entité. Les raisons de cette inaccessibilité peuvent être dues à plusieurs facteurs : handicap, restrictions administratives ou indisponibilité de moyen de déplacement. Par conséquent, certaines coordonnées peuvent ne pas être prises en compte, ce qui améliore l'exactitude et influence les décisions [Pfeifer, 2005].

## 2.2. DÉFINITIONS

---



FIGURE 2.1 – Étude du comportement des vaches par WSN. Source [Nadimi *et al.*, 2007]



FIGURE 2.2 – Positionnement d'une personne à l'intérieur d'un bâtiment. Source [Fraunhofer IIS, 2013]



FIGURE 2.3 – Positionnement de robot humanoïde à l'intérieur d'un bâtiment. Source [Hornung *et al.*, 2010].

En définitive, une position est plus que des coordonnées géographiques. Elle permet en fonction du modèle choisi pour l'application de définir une rue, une salle de conférence, de traduire toutes coordonnées géographiques en nom de chambre, etc. Plusieurs modèles sont décrits dans [Mabrouk *et al.*, 2005].

Dans [Pfeifer, 2005], la position a été découpée en quatre niveaux de granularité : i) Position (un point sur un plan pour une entité, absolue ou relative) ; ii) une adresse (description structurée et textuelle) décrivant comment trouver la Position ; iii) Zone Géographique et iv) Structure (Représentant les immeubles, les étages, les salles et les autres entités).

Une entité peut être une personne, un animal ou un objet portant ou non un appareil spécial appelé nœud dans la suite. De ce fait, dans la majorité des cas l'entité est localisée grâce à ce nœud. La Figure 2.2 illustre le positionnement d'une vache, d'une personne et d'un humanoïde.

Les nœuds disposant de positions connues sont appelés *Anchor Nodes*, *Beacons*, *Fix-points*, *Access Points (AP)*, *Base Stations* ou *Reference Nodes*. Ces termes renvoient à des types de nœuds et des méthodes particulières utilisées par les nœuds et le système pour obtenir les informations de localisation. Les nœuds sont représentés physiquement par des marqueurs ou des nœuds à des positions permanentes et sont fixés à certaines positions de coordonnées connues.

Par contraste, les nœuds dont les positions sont inconnues sont dénotés par *Mobile Stations*, *Mobile Terminal*, *Blind Nodes* et doivent être positionnés par le système de localisation. La plupart des systèmes de positionnement prennent en compte la mobilité des nœuds. Ces nœuds sont attachés aux utilisateurs du système sous la forme de tags ou d'autres types d'équipements matériels ayant des capacités de communication et de traitement. Il peut aussi être un dispositif couplé à un robot. [Mautz, 2009]

**La localisation (*Localization*)** est plus utilisée pour dénoter le positionnement d'un nœud par réseaux de capteurs, lorsque ces derniers sont vus comme des nœuds communicants de façon *ad-hoc* et coopérative en fournissant une estimation de la position surtout par positionnement collaboratif. Nous allons utiliser ce terme pour indiquer simplement le positionnement d'une entité dans un milieu donné.

**La géolocalisation ou géoréférencement (*Geolocation*)**, est définie comme étant un procédé permettant de positionner une entité sur un plan ou une carte à l'aide de ses coordonnées géographiques. Celles-ci sont obtenues à l'aide d'un système de positionnement

par satellites, réseaux cellulaires (GSM, LTE), WiFi, ou des Web services, entre autre technologies [Gentile *et al.*, 2012, Schmidt-Dannert et SNET, 2010]. Le système localise le nœud porté par l'entité et celui-ci peut stocker, pré-traiter les informations de localisation avant de les envoyer par liaison sans fil ou non à une plateforme logicielle de géolocalisation.

**La navigation (*Navigation*)** est un service basé sur la position. Elle se sert de la position courante, de la vitesse et de la direction d'une entité fournies par le système pour : i) déterminer le chemin optimal à partir de la position courante par optimisation mono-critère (plus court, moins coûteux) ou multicritères (combinaison de critères) ; ii) guider suivant un chemin donné et contrôler l'écart entre la position courante et le chemin planifié. La navigation requiert des informations sur l'espace navigable. L'espace navigable est défini par rapport à des contraintes liées aux lieux restreints, par exemple nécessitant des autorisations et aux caractéristiques de l'utilisateur, personnes handicapées par exemple [Mautz, 2012].

Par contraste au *tracking* qui est la détermination de la position d'une entité en mouvement de façon répétée dans le temps par une infrastructure, la navigation requiert que les informations de positionnement soient disponibles sur le nœud mobile.

**Le positionnement (*Positioning*)** est le terme général utilisé pour désigner la détermination de la position d'une entité portant ou non un équipement particulier (téléphone portable, ardoise, capteur, puce RFID, etc). Le positionnement peut se dérouler de deux façons : centralisée ou distribuée.

Le concept d'une architecture centralisée est que toute la procédure de positionnement se déroule au niveau d'un serveur central où toutes les positions sont stockées et accessibles à une tierce partie (administrateurs, clients, etc.). Les avantages d'une telle architecture sont la simplicité, une base de données commune, un service uniforme à tous les utilisateurs et un faible coût. Par contre, pour un système distribué, la détermination de la position se fait sur chaque nœud de façon locale en se basant sur des observations locales. Les avantages sont une grande évolutivité et la préservation de la vie privée des utilisateurs.

Nous pouvons aussi distinguer entre système de localisation local (*Local Positioning System* LPS) et système de localisation temps réel. Un LPS, de façon similaire au positionnement global de type GNSS, utilise une infrastructure locale pour positionner une entité. La position locale obtenue est relative au repère défini par l'infrastructure et pourrait ne pas être exploitable à l'extérieur de ce repère. Un RTLS est un système de positionnement permettant de fournir la position en respectant une contrainte temporelle allant jusqu'à l'instantanée [Malik, 2009].

Dans la plupart des cas, les systèmes de positionnement se composent de trois étapes : i) une phase de mesures ; ii) la détermination de la position ainsi que la modélisation et la prédiction du comportement (en terme de trajectoire, place à ne pas visiter, etc.) de l'entité à localiser ; iii) optionnellement une phase de raffinement de la position. Ce processus est illustré par la figure 2.4.

## 2.3 Les techniques de bases de mesures

Le positionnement d'une entité tire parti des *informations obtenues de ou sur son environnement*. La diversité avec laquelle ces informations peuvent être obtenues, rend compte

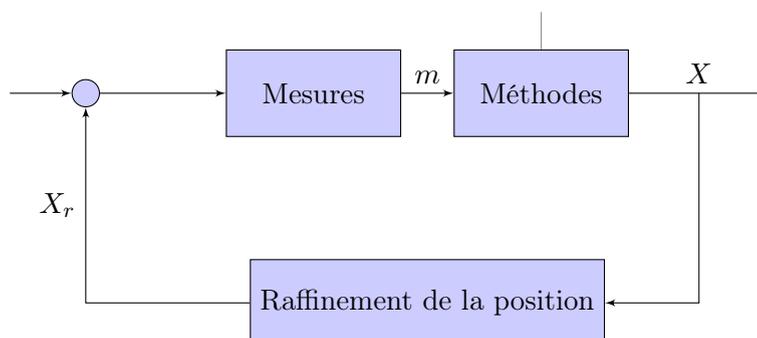


FIGURE 2.4 – Cette figure décrit le processus de la localisation. Qui consiste en la mesure d’une ou de plusieurs grandeurs physiques à l’aide de techniques de mesures. Le résultat de cette première phase est symbolisé par  $m$ , qui est combiné à des connaissances a priori sur les positions de points de référence pour permettre à une méthode de positionnement de déterminer la position  $X$ . Cette position peut optionnellement être raffinée. Le raffinement est un processus itératif qui permet de minimiser l’erreur de positionnement jusqu’à un certain seuil. La position  $X_r$  est alors fournie à l’entité.

des nombreux travaux effectués et en cours sur le positionnement des entités. Par exemple, le changement des conditions ambiantes (température, luminosité, présence/absence), le contact sur une surface capacitive, une parole peuvent permettre de positionner une entité. Des approches existent pour la fusion de ces informations [Pfeifer, 2005]. Néanmoins dans le cadre de cette thèse, nous allons utiliser les caractéristiques de l’onde électromagnétique pour la localisation *indoor*. Ces caractéristiques sont : l’amplitude du signal, l’angle ou la direction d’arrivée et le temps d’arrivée.

Dans cette section nous allons présenter les techniques les plus utilisées pour extraire ces caractéristiques d’un signal envoyé ou reçu.

### 2.3.1 La Mesure de la Puissance du Signal

La puissance d’un signal décroît, selon le modèle de Friis [Rappaport, 2002], proportionnellement à  $1/d^\eta$  entre un émetteur et un récepteur, où  $d$  est la distance qui les séparent et  $\eta$  est l’exposant de perte dépendant du milieu de propagation. Les difficultés principales que l’implémentation d’une solution de localisation à base de la mesure de la puissance du signal ou *Received Signal Strength Indicator* (RSSI) rencontrent sont les perturbations dans l’environnement. Bien que des modèles existent pour divers environnements, ils nécessitent un paramétrage fin [Hashemi, 1993, Sarkar *et al.*, 2003, Agrawal et Patwari, 2009].

Un des modèles les plus utilisés et dérivant du modèle de Friis, est le suivant :

$$P_r(d) = P_0(d_0) - \eta 10 \log_{10} \left( \frac{d}{d_0} \right) + X_\sigma \quad (2.1)$$

$P_r(d)$  dénote la puissance du signal reçu à la distance  $d$  et  $P_0(d_0)$  est la puissance reçue à une certaine distance de référence  $d_0$ ,  $\eta$  est l’exposant de perte du chemin et

$X_\sigma$  est la variable aléatoire log-normal avec une variance  $\sigma^2$  qui mesure les effets de masque. Si l'exposant de perte est connu pour un environnement, la puissance du signal reçu peut être utilisée pour estimer la distance. Des valeurs pour la variable aléatoire,  $X_\sigma$ , sont données dans [Rappaport, 2002, Simon et Alouini, 2004] et aussi dans le manuel [Fall et Varadhan, 2002] du simulateur NS-2.

Un estimateur de maximum de vraisemblance de la distance  $d$  peut être déterminé comme suit :

$$\hat{d} = d_0 \left( \frac{P_r}{P_0(d_0)} \right)^{-1/\eta} \quad (2.2)$$

Cet estimateur de maximum de vraisemblance est biaisé. Un estimateur non biaisé est [Mao *et al.*, 2007] :

$$\hat{d} = d_0 \left( \frac{P_r}{P_0(d_0)} \right)^{-1/\eta} e^{-(\alpha/2)(\sigma^2/\eta^2)} \quad (2.3)$$

Pour cette technique, et les autres que nous allons décrire, il existe une limite inférieure à cet estimateur qui est donnée par la borne de Cramer et Rao [Gezici *et al.*, 2005].

La mesure de la puissance du signal reste néanmoins la technique de mesure de choix car plus accessible et facile à mettre en œuvre. Cependant, elle dépend des technologies utilisées.

Le problème consiste alors à estimer  $x$  sachant que :  $r = [\tilde{d}_1, \dots, \tilde{d}_N]^T$  les estimations des distances,  $[x_1, \dots, x_N]$  les positions connues des nœuds de référence et  $f(x) = [f_1(x), \dots, f_N(x)]$ .<sup>2</sup> Des méthodes dites géométriques sont utilisées pour déterminer la position [Torrieri, 1984].

### 2.3.2 Le temps d'arrivée

La distance moyenne parcourue entre un point  $A$  à l'instant  $t_1$  et un point  $B$  à l'instant  $t_2$  est donnée par  $d = v\Delta t$ , où  $v$  est la vitesse moyenne et  $\Delta t = t_2 - t_1$ . Ainsi, lorsque la vitesse est connue, l'estimation de  $\Delta t$  permet de déterminer la distance. Deux catégories de mesures du temps de propagation d'un signal sont couramment utilisées [Gezici *et al.*, 2005, Gustafsson et Gunnarsson, 2005] : les techniques dites *one-way* et celles dites *round trip propagation*. Ces deux techniques sont illustrées par la figure 2.5.

Dans le premier cas, l'émetteur envoie un message contenant le temps d'envoi  $t_1$  et au moment de la réception le récepteur enregistre le temps de réception  $t_2$  puis détermine  $\Delta t$ . En général, c'est la vitesse de la lumière qui est utilisée. Cette méthode nécessite une synchronisation des horloges, ce qui introduit un coût au niveau du nœud et un coût en transmission réseau.

La seconde catégorie permet de déterminer le temps mis par le signal pour aller et revenir d'un émetteur vers un récepteur sans synchroniser les horloges. Dans ce cas, l'émetteur enregistre aussi, en plus du temps d'envoi, le temps d'attente, ce que fera aussi le récepteur

---

2.  $f_i(x) = d^2(x, x_i) = (x - x_i)^2 + (y - y_i)^2$  dans le cas d'une distance euclidienne.

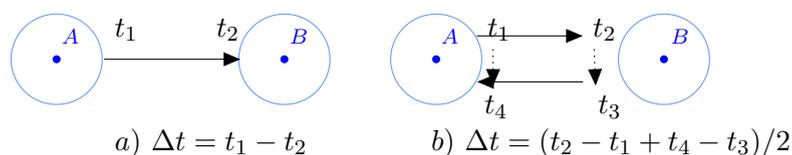


FIGURE 2.5 – Illustration des deux versions de la mesure du temps d’arrivée d’un signal. Dans le cas a), le temps de propagation  $\Delta t(t_2 - t_1)$  est calculé intuitivement, mais requiert une synchronisation et l’envoi d’une estampille. Cependant le temps d’aller-retour, illustré par le cas b), permet de relâcher la contrainte de synchronisation et on peut en trouver plusieurs déclinaisons en fonction de la gestion des temps de traitement sur les nœuds.

qui enregistre le temps de réception et de traitement du message. C’est la différence de ces temps qui donnera le temps d’arrivée.

Plusieurs variantes de ces techniques de mesure existent avec une contrainte plus ou moins relâchée sur la synchronisation et requérant des matériels spécifiques [Liu *et al.*, 2010]. *Symmetric Double Sided Two Way Ranging* (SDS-TWR) en est un exemple qui répète l’aller-retour deux fois [Jiang et Leung, 2007]. Bien que ces deux catégories de techniques peuvent être appliquées dans plusieurs normes de réseaux filaires ou sans fil, l’*Ultra Wide Band* offre de meilleurs résultats en raison du signal en ligne direct relativement facile à déterminer [Gezici *et al.*, 2005].

Après la détermination des temps, des distances peuvent être calculées et le problème formulé comme pour RSSI. La précision de ces techniques dépend fortement des horloges et aussi de l’environnement.

### 2.3.3 L’angle d’arrivée

Les lignes de portée d’un signal reçu par plusieurs récepteurs (deux au minimum) s’entrecroisent à la position de l’émetteur lorsqu’il n’y a pas d’erreurs dans les mesures, comme illustré à la figure 2.6.a. Cependant, en cas d’erreurs, les lignes ne s’entrecroisent pas [Torrieri, 1984].

- La formulation du problème de localisation dans ce cas devient, déterminer  $x$ , lorsque :
- $r = [\beta_1, \dots, \beta_N]^T$  : les mesures des portées ;
  - $N$  : est le nombre total de récepteurs ;
  - $x_i = [x_i, y_i], 1 \leq i \leq N$  : les positions connues des nœuds de référence ;
  - $\phi(x) = [\phi_1, \dots, \phi_N]^T$  : les vraies valeurs des portées inconnues ;
  - $f(x) = \tan \phi(x)$  : la relation entre la position du récepteur et celle de l’émetteur.

Les techniques AOA (*Angle of Arrival*) sont très dépendantes, en terme de précision, de la sensibilité des récepteurs, de la largeur du faisceau utilisée pour estimer l’angle, de la directivité de l’antenne et des problèmes de multi-trajets. Elles ajoutent aussi un surcoût en requérant un dispositif spécial pour la mesure de l’angle. Néanmoins, elles obtiennent de meilleurs résultats comparées aux techniques basées sur la puissance du signal [Stansfield, 1947].

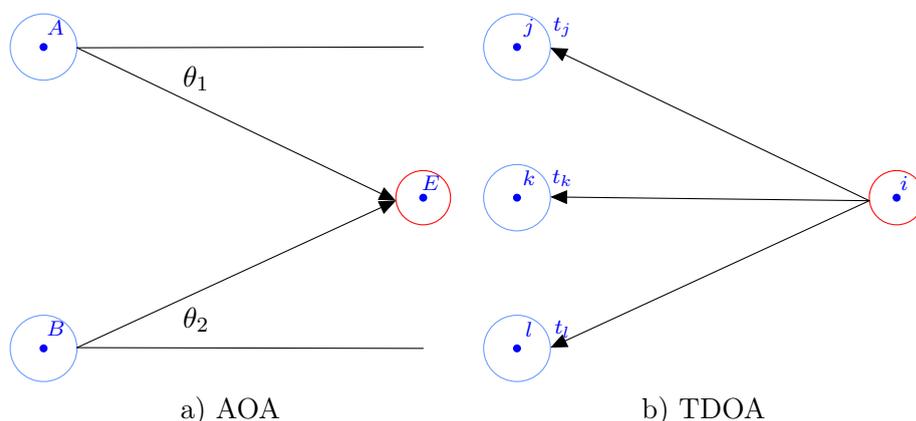


FIGURE 2.6 – Illustration des techniques de mesures Angle d’Arrivée et la différence entre temps d’arrivée. Dans le cas du AOA, la position du nœud est donnée par l’intersection des lignes de portée des nœuds  $A$  et  $B$ . Dans le cas du TDOA, le temps de réception au niveau des récepteurs  $j, k, l$  permet de déterminer la position de  $i$ .

### 2.3.4 La Différence entre Temps d’Arrivée

La position d’un émetteur peut être estimée grâce aux différences entre les temps de réception, sur divers récepteurs, du signal qu’il a émis, comme illustré à la figure 2.6.b. Considérons un émetteur  $i$  et deux récepteurs  $j$  et  $k$ , la différence  $\Delta t_{jk}$  entre les temps d’arrivée du signal à l’émetteur  $j$  et  $k$  est  $\Delta t_{jk} = t_j - t_k = c^{-1}(d(x_j, x_i) - d(x_k, x_i))$ , où  $t_j$  et  $t_k$  sont les temps de réception du signal sur  $j$  et  $k$  et  $d(.,.)$  est la distance euclidienne et  $c$  est la vitesse de la lumière. La position de  $i$  se retrouve à l’intersection des hyperboles dont les foyers se trouvent aux positions de  $j$  et  $k$ . Torrieri en a donné une description plus complète [Torrieri, 1984].

Une différence est à faire entre le TDOA (*Time Difference Of Arrival*) défini ici, et la définition donnée par les auteurs du système Cricket [Priyantha *et al.*, 2000]. Dans ce système, le TDOA est la différence entre un signal plus rapide se propageant à la vitesse de la lumière et un signal ultrason.

Les techniques présentées dans les sous-sections précédentes souffrent, en plus du multi-trajet, du *Geometric Dillution of Precision (GDOP)* où une petite erreur de mesure peut résulter en une grande erreur d’estimation [Mao *et al.*, 2007].

### 2.3.5 Le Profilage de la Puissance du Signal

Cette technique est basée sur la construction d’une empreinte unique de la propagation du signal dans un environnement. Le profilage se nomme aussi *fingerprinting* car elle consiste à construire un *fingerprint*. Le terme *fingerprinting* sert aussi à désigner les algorithmes de positionnement se servant du profilage.

Ainsi, l’hypothèse est que cette empreinte soit un identifiant pour le milieu dans lequel elle est prise. L’empreinte est obtenue, le plus souvent, en mesurant la puissance du signal pour plusieurs points et en considérant les positions des nœuds de référence. On

obtient alors un ensemble de vecteurs, dont certaines entrées peuvent être nulles à cause de l'éloignement du nœud de référence correspondant.

Soit  $(x, y)$  le point en question, et  $rss_i(x, y)$  la puissance du signal provenant du point d'accès  $i$ . L'ensemble des mesures pour un point donné est :

$$rss(x, y) = [rss_i(x, y), i = 1..n]$$

Avec  $n$  le nombre de points d'accès.

Toutes ces mesures sont conservées dans une base de données. Une dépendance spatio-temporelle est à prendre en compte dans la mise en place du *fingerprint* du fait des effets de masque et des évanouissements non sélectifs (voir Chapitre 1). Deux cas de figure se présentent : la détermination d'une mesure de tendance centrale, ou la caractérisation des signaux reçus pour chaque point par une distribution de probabilité.

La moyenne est la mesure de tendance centrale la plus couramment utilisée. Il s'agit de simplement conserver dans la base de données, pour  $(x, y)$ , la moyenne de toutes les puissances reçues. Ainsi :

$$r\bar{ss}(x, y) = [r\bar{ss}_i(x, y), i = 1..n] \quad (2.4)$$

$$r\bar{ss}_i(x, y) = \frac{1}{m} \sum_j rss_j(x, y) \quad (2.5)$$

Pour un nombre de mesures  $m$  effectuées pour la position  $(x, y)$ .  $n$  est la taille du vecteur. Les inconvénients d'une telle approche sont la perte d'information et aussi la sensibilité aux valeurs aberrantes dans le cas de la moyenne. D'autres approches consistent à utiliser la médiane ou le mode.

La modélisation de la puissance du signal reçu par une distribution de probabilité est plus réaliste et permet un meilleur positionnement car il permet de prendre en compte les fluctuations instantanées du canal radio [Evennou, 2007]. Cependant, il nécessite qu'un grand nombre de mesures soit effectué pour déterminer les paramètres de la loi.

Pour estimer sa position, le nœud mesure le vecteur de puissance correspondant à sa position courante et ce vecteur est comparé à la base de données des mesures pour déterminer le ou les vecteurs les plus proches. La position du nœud est alors celle associée à ce vecteur.

Ici, le problème de la localisation peut être reformulé en problème de classification supervisée. Les réseaux de neurones artificiels, les réseaux bayésiens peuvent être utilisés pour le résoudre [Mao *et al.*, 2007].

## 2.4 Les méthodes de positionnement

Une fois les informations sur l'environnement obtenues, les méthodes de positionnement permettent de déterminer la position. Ces méthodes peuvent être classifiées suivant plusieurs critères [Bejuri *et al.*, 2011]. Nous allons présenter les méthodes dites basées sur la mesure, qui utilisent les techniques présentées dans la section 2.3. Ces méthodes sont basées

sur l'hypothèse que le nœud à localiser ait accès à suffisamment de mesures provenant de suffisamment de stations dont les positions sont connues.

Supposons qu'un nœud reçoive un ensemble de signaux provenant de  $n$  stations de base placées à des positions connues  $x_i, i = 1, \dots, n$  dans la zone de déploiement. Dans le cas où les mesures  $m_i, i = 1, \dots, n$  ne comportent pas d'erreurs, on a :  $m_i = f_i(x)$  où en général  $f_i(x)$  est une fonction non-linéaire [Torrieri, 1984]. Mais, les mesures comportent en général des erreurs additives  $e_i$ , ce qui transforme la relation en :

$$m_i = f_i(x) + e_i, i = 1, \dots, N \quad (2.6)$$

Ces  $n$  équations sont résumées en une équation vectorielle à  $n$ -dimensions :

$$m = f(x) + e \quad (2.7)$$

$m$  représente la quantité mesurée (ou estimée) généralement une distance,  $f_i(x)$  est la fonction qui relie la position inconnue à  $i$ ème position connue et  $e$  est l'erreur qui est supposée suivre une distribution Gaussienne de moyenne nulle :

$$p_e(e) = N(0, \sigma)$$

Un cas simple et efficient, suivant la distribution de probabilité de l'erreur, consiste à déterminer un estimateur  $\hat{x}$  qui est :

- un Estimateur de Maximum de Vraisemblance, lorsque l'erreur est gaussienne ;
- un Estimateur des Moindres Carrés dans le cas non gaussien.

Cependant, lorsque cette hypothèse n'est pas vérifiée, les nœuds peuvent collaborer entre eux pour se localiser. C'est le cas dans les réseaux de capteurs [Patwari *et al.*, 2005] : c'est la localisation collaborative.

Certains auteurs considèrent dès le départ que tout ou la majorité des nœuds ne savent pas leurs positions et ont mis en place des algorithmes distribués [Bulusu *et al.*, 2000, Savarese *et al.*, 2001, Savvides *et al.*, 2001, Niculescu et Nath, 2003, He *et al.*, 2003].

Le problème de la localisation à l'intérieur de bâtiment peut aussi être modélisé sous la forme d'un problème d'optimisation ou de classification non supervisée et, de ce fait, plusieurs algorithmes ont été utilisés depuis Doherty [Doherty *et al.*, 2001] jusqu'aux méthodes bio-inspirées : PSO [Kulkarni et Kumar, 2011] par exemple.

Nous allons présenter les méthodes de positionnement suivantes :

- les méthodes géométriques ;
- les méthodes à minimisation d'une fonction coût ;
- les méthodes de fingerprinting ;
- les techniques bayésiennes ;

### 2.4.1 Les méthodes géométriques

Les méthodes géométriques sont ainsi nommées parce que la relation qui lie la position inconnue  $x$  aux autres positions est géométrique. On retrouve aussi, dans la littérature, ces méthodes sous la dénomination de : « triangulation, trilatération et multilatération ». Ces méthodes sont illustrées par la figure 2.7.

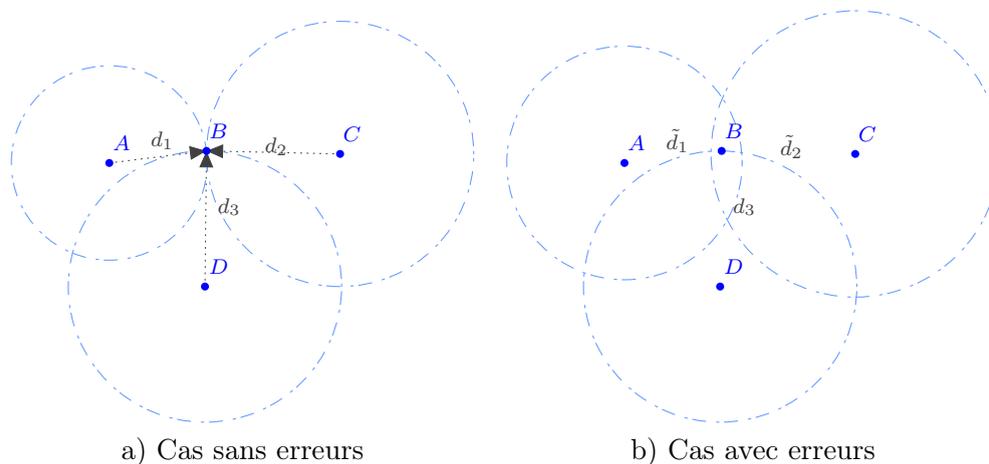


FIGURE 2.7 – a) L’intersection des trois cercles donne la position du nœud. Cette méthode est largement utilisée lorsque la distance euclidienne est estimée ou calculée. b) Avec des erreurs de mesures.

La triangulation est une méthode très ancienne qui détermine les coordonnées d’un des sommets d’un triangle connaissant les mesures des angles et des coordonnées des autres sommets.

Ainsi dans le cas sans erreur, trois cercles, trois hyperboles, deux lignes de portées se coupent en un point.

En considérant que la quantité mesurée est une distance, nous avons ces  $n$  équations :

$$\tilde{d}_i = d_i + e_i$$

Avec  $\tilde{d}_i$  la distance estimée et séparant la position courante du point de référence  $i$ , tandis que  $d_i$  est la vraie distance. Par souci de simplicité de notation, nous allons utiliser  $d_i$  dans nos équations à la place de  $\tilde{d}$ .

Plusieurs méthodes de résolution peuvent être appliquées pour résoudre  $m$ . Nous allons détailler certaines.

Une d’entre elles consiste en la soustraction de la première équation des  $n$  autres, de la deuxième des  $n - 1$  et ainsi de suite. Ce qui nous donne une équation linéaire de forme :  $R = AX$ . Dans le cas où  $n = 3$ , la position peut être facilement déterminée.

### 2.4.1.1 Méthode de soustraction

La méthode simple de trilatération consiste à déterminer la solution d’un système d’équations linéaires à  $D$  inconnues, avec  $D$  la dimension du domaine de positionnement. Le système d’équations est déterminé suite à une série de soustractions.

Soient les distances mesurées  $d_i$ , avec  $i = 1..3$  alors nous avons :

$$d_i^2 = (x - x_i)^2 + (y - y_i)^2 \tag{2.8}$$

## 2.4. LES MÉTHODES DE POSITIONNEMENT

---

Avec  $(x, y)$  les coordonnées en 2D du nœud mobile et  $(x_i, y_i)$  celles du nœud de référence  $i$ . En développant ces 3 équations nous avons :

$$d_1^2 = x^2 - 2x.x_1 + x_1^2 + y^2 - 2y.y_1 + y_1^2 \quad (2.9)$$

$$d_2^2 = x^2 - 2x.x_2 + x_2^2 + y^2 - 2y.y_2 + y_2^2 \quad (2.10)$$

$$d_3^2 = x^2 - 2x.x_3 + x_3^2 + y^2 - 2y.y_3 + y_3^2 \quad (2.11)$$

En soustrayant  $d_2$  et  $d_3$  de  $d_1$  nous avons :

$$d_1^2 - d_2^2 = -2x(x_1 - x_2) + (x_1^2 - x_2^2) + (y_1^2 - y_2^2) - 2y(y_1 - y_2) \quad (2.12)$$

$$d_1^2 - d_3^2 = -2x(x_1 - x_3) + (x_1^2 - x_3^2) + (y_1^2 - y_3^2) - 2y(y_1 - y_3) \quad (2.13)$$

Soit :

$$- \delta x_{ij} = x_i - x_j,$$

$$- \delta y_{ij} = y_i - y_j,$$

$$- \delta z_{ij} = z_i - z_j,$$

En résolvant ce système d'équations, la position est indiquée par :

$$\hat{y} = \frac{\delta x_{13} [d_1^2 - d_2^2 - (x_1^2 - x_2^2) - (y_1^2 - y_2^2)] - \delta x_{12} [d_1^2 - d_3^2 - (x_1^2 - x_3^2) - (y_1^2 - y_3^2)]}{2\delta y_{13}\delta x_{12} - 2\delta y_{12}\delta x_{13}} \quad (2.14)$$

$$\hat{x} = \frac{d_1^2 - d_2^2 + (x_1^2 - x_2^2) + (y_1^2 - y_2^2) + 2\hat{y}\delta y_{12}}{-2\delta x_{12}} \quad (2.15)$$

De façon générale, ce système d'équations peut s'exprimer pour  $n$  équations sous une forme matricielle.

$$R = A.X \quad (2.16)$$

Avec :

$$R = \begin{bmatrix} d_2^2 - d_1^2 - \|X_1\|^2 - \|X_2\|^2 \\ \vdots \\ d_n^2 - d_1^2 - \|X_1\|^2 - \|X_n\|^2 \end{bmatrix}, A = \begin{bmatrix} 2(x_2 - x_1)2(y_2 - y_1) \\ \vdots \\ 2(x_n - x_1)2(y_n - y_1) \end{bmatrix}, X = \begin{bmatrix} x \\ y \end{bmatrix}$$

Si la matrice  $A$  est non singulière alors une solution est obtenue par  $X = A^{-1}R$ . Une matrice non singulière est une matrice inversible.

Les avantages d'une telle méthode est la simplicité et la rapidité d'implémentation. Nous allons utiliser les résultats fournis par cette méthode comme entrée à d'autres méthodes.

### 2.4.1.2 La méthode directe

La méthode directe (Direct Method) est une autre méthode simple permettant de déterminer une position après linéarisation. En 3D, elle retourne deux positions en fonction des valeurs estimées de  $z$ .

## 2.4. LES MÉTHODES DE POSITIONNEMENT

---

Avec  $(x, y, z)$  les coordonnées en 3D du nœud mobile et  $(x_i, y_i, z_i)$  celles du nœud de référence  $i$ .

$$d_i = \sqrt{(x_i^2 - x)^2 + (y_i^2 - y)^2 + (z_i^2 - z)^2} \quad (2.17)$$

En développant cette équation nous avons :

$$d_i^2 = x^2 - 2x.x_i + x_i^2 + y^2 - 2y.y_i + y_i^2 + z^2 - 2z.z_i + z_i^2 \quad (2.18)$$

En procédant par différence, comme pour le cas précédent, nous posons :

$$— \delta d_{ij} = ((x_i^2 + y_i^2 + z_i^2) - (x_j^2 + y_j^2 + z_j^2) + d_i^2 - d_j^2)/2$$

La méthode DM retourne les solutions suivantes :

$$\hat{z} = \frac{A}{B} \pm \sqrt{\left(\frac{B}{A}\right)^2 - \frac{C}{A}} \quad (2.19)$$

$$\hat{x} = \gamma \hat{z} + \mu \quad (2.20)$$

$$\hat{y} = \alpha \hat{z} + \beta \quad (2.21)$$

Avec :

$$A = \alpha^2 + \beta^2 + 1; \quad (2.22)$$

$$B = \alpha(y_1 - \beta) + \gamma(x_1 - \mu) + z_1; \quad (2.23)$$

$$C = (x_1 - \mu)^2 + (y_1 - \beta)^2 + z_1^2 - d_1^2, \quad (2.24)$$

$$\alpha = \frac{\delta x_{21} \delta z_{31} - \delta x_{31} \delta z_{21}}{\delta x_{31} \delta y_{21} - \delta x_{21} \delta y_{31}} \quad (2.25)$$

$$\beta = \frac{\delta x_{31} \delta d_{21} - \delta x_{21} \delta d_{31}}{\delta x_{3,1} \delta y_{21} - \delta x_{21} \delta y_{31}} \quad (2.26)$$

$$\gamma = \frac{\delta y_{31} \delta z_{21} - \delta x_{31} \delta z_{21}}{\delta x_{31} \delta y_{21} - \delta x_{21} \delta y_{31}} \quad (2.27)$$

$$\mu = \frac{\delta y_{21} \delta d_{31} - \delta y_{31} \delta d_{21}}{\delta x_{31} \delta y_{21} - \delta x_{21} \delta y_{31}} \quad (2.28)$$

Cette méthode retourne deux solutions en fonction des deux valeurs de  $\hat{z}$ . L'intérêt en est la possibilité d'utiliser des informations a priori pour choisir une des deux valeurs.

Lorsque  $z$  est nul,  $\alpha, \gamma$  et  $B$  le sont aussi. DM revient alors au cas de figure traité dans la section précédente.

D'autres méthodes dites directes ont été présentées dans [Yan, 2010].

### 2.4.1.3 La méthode de Chan

Cette méthode permet de résoudre l'équation hyperbolique obtenue par mesure du TDOA [Chan et Ho, 1994]. En considérant le cas de quatre nœuds de référence on a :

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = - \begin{bmatrix} \delta x_{21} & \delta y_{21} & \delta z_{21} \\ \delta x_{31} & \delta y_{31} & \delta z_{31} \\ \delta x_{41} & \delta y_{41} & \delta z_{41} \end{bmatrix}^{-1} \left\{ \begin{bmatrix} d_{21} \\ d_{31} \\ d_{41} \end{bmatrix} \times R_1 + 0.5 \times \begin{bmatrix} d_{21}^2 - k_2 + k_1 \\ d_{31}^2 - k_3 + k_1 \\ d_{41}^2 - k_4 + k_1 \end{bmatrix} \right\} \quad (2.29)$$

Avec  $k_1 = x_i^2 + y_i^2 + z_i^2$ , et  $i = 1..4$ .

La méthode de Chan est une méthode approchée de la méthodes des moindres carrés qui selon les auteurs donne de meilleurs résultats en terme du CRLB que la méthode par série de Taylor, la méthode du Divide and Conquer (DAC). Les détails de ces méthodes sont aussi données dans [Chan et Ho, 1994].

#### 2.4.1.4 La méthode des moindres carrés

En reprenant notre système d'équation  $AX = R$ , la méthode des moindres carrés permet de déterminer itérativement :

$$\hat{X} = (A^T A)^{-1} A R \quad (2.30)$$

L'application de la méthode des moindres carrés nécessite une position initiale  $X = [x, y]^T$ .

Nous redéfinissons par la suite l'équation 2.30 de la façon suivante :

$$\begin{bmatrix} \delta d_1 \\ \vdots \\ \delta d_n \end{bmatrix} = \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \vdots & \vdots \\ \alpha_{n1} & \alpha_{n2} \end{bmatrix} \begin{bmatrix} \delta x \\ \delta y \end{bmatrix} \quad (2.31)$$

Avec :

$$d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2} \quad (2.32)$$

$$\delta d_i = \frac{(x - x_i)\delta x + (y - y_i)\delta y}{d_i} \quad (2.33)$$

$$\alpha_{i1} = \frac{x - x_i}{d_i} \quad \alpha_{i2} = \frac{y - y_i}{d_i} \quad (2.34)$$

Ainsi, le système d'équations se réécrit en :

$$\delta d = \alpha \cdot \delta x \quad (2.35)$$

Dans certains cas exceptionnels, la matrice ne peut pas être inversée et le calcul échoue [Evennou, 2007]. L'algorithme des moindres carrés pour la localisation est données par l'algorithme 1.

#### 2.4.1.5 Les séries de Taylor

Cette consiste à obtenir des estimateurs à partir du développement limité du système d'équations. Le développement par Série de Taylor est défini pour une fonction  $f(x)$  en un point  $x_0$  par :

$$\sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!} (x - x_0)^n \quad (2.36)$$

---

**Algorithme 1 : Moindres carrés**

---

- 1: Initialisation de l'algorithme  $x \leftarrow x_0$
  - 2: **repeat**
  - 3: Déterminer les pseudo-distances  $d_i$
  - 4: Déterminer les coefficients de la matrice  $\alpha$
  - 5:  $\delta x \leftarrow [\alpha^T \cdot \alpha]^{-1} \cdot \alpha^T \cdot \delta d$
  - 6:  $\delta v \leftarrow \sqrt{\delta x^2 + \delta y^2}$
  - 7: **if**  $\delta < Seuil$  **then**
  - 8:      $x \leftarrow x + \delta x$
  - 9: **end if**
  - 10: **until**  $\delta v > Seuil$
- 

Où  $n!$  est la factorielle de  $n$  et  $f^{(n)}(x_0)$  est la dérivée n-ième de la fonction  $f$ .

L'application de la linéarisation par série de Taylor au positionnement se déroule de la façon suivante.

Soit la formulation du problème de positionnement en 2D :

$$f(x, y) = \begin{bmatrix} d_1 - \sqrt{(x - x_1)^2 + (y - y_1)^2} \\ d_2 - \sqrt{(x - x_2)^2 + (y - y_2)^2} \\ d_3 - \sqrt{(x - x_3)^2 + (y - y_3)^2} \end{bmatrix} \quad (2.37)$$

Le jacobien de cette fonction est donné par :

$$J_f = \begin{bmatrix} \frac{\delta f_1}{\delta x} & \frac{\delta f_1}{\delta y} \\ \frac{\delta f_2}{\delta x} & \frac{\delta f_2}{\delta y} \\ \frac{\delta f_3}{\delta x} & \frac{\delta f_3}{\delta y} \end{bmatrix} = \begin{bmatrix} \frac{(x-x_1)}{d_1 - \sqrt{(x-x_1)^2 + (y-y_1)^2}} & \frac{(y-y_1)}{d_1 - \sqrt{(x-x_1)^2 + (y-y_1)^2}} \\ \frac{(x-x_2)}{d_2 - \sqrt{(x-x_2)^2 + (y-y_2)^2}} & \frac{(y-y_2)}{d_2 - \sqrt{(x-x_2)^2 + (y-y_2)^2}} \\ \frac{(x-x_3)}{d_3 - \sqrt{(x-x_3)^2 + (y-y_3)^2}} & \frac{(y-y_3)}{d_3 - \sqrt{(x-x_3)^2 + (y-y_3)^2}} \end{bmatrix} \quad (2.38)$$

$J_f$  est utilisé de la façon suivante pour approximer la fonction  $f(x, y)$  :

$$f(x, y) = f(x_0, y_0) + J_f \times (X - X_0) \quad (2.39)$$

Avec  $X = (x, y)$  et  $X_0 = (x_0, y_0)$  une position de départ.

À partir de cette équation, la position peut être améliorée de façon itérative.

Les méthodes itératives, série de Taylor et moindres carrés peuvent rencontrer des problèmes de convergence. En effet la position initiale influence largement cette convergence. Cependant, elles ont l'avantage d'être faciles et simples à mettre en œuvre.

#### 2.4.1.6 Min-Max

La méthode Min-Max construit des carrés autour des positions des nœuds de référence à partir des distances mesurées. La figure 2.8 illustre ce cas pour trois nœuds de référence. Pour un nœud de référence  $i$ , le carré est construit de la façon suivante à partir de ses

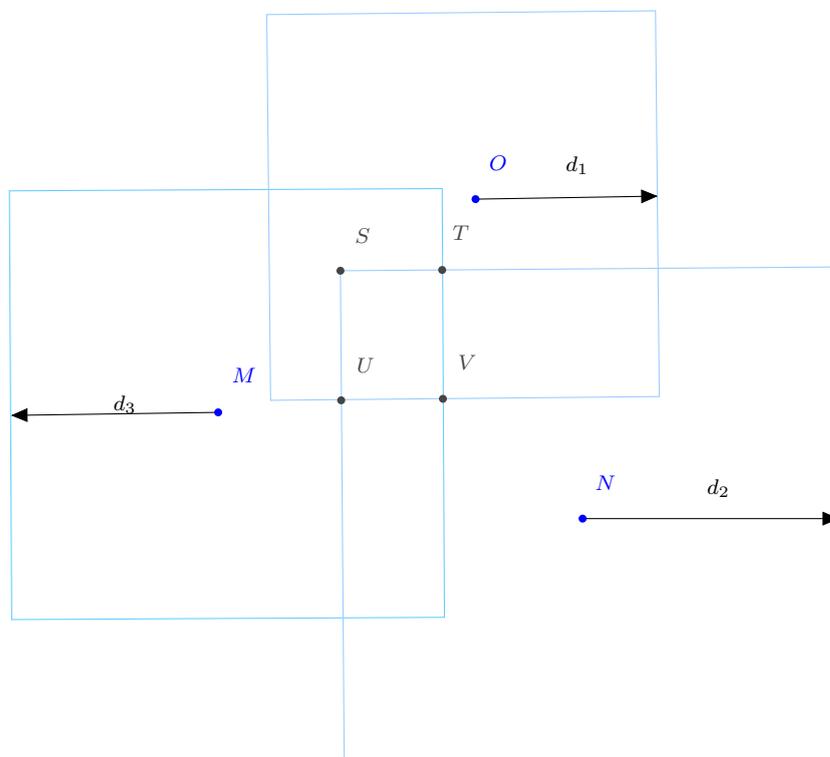


FIGURE 2.8 – Illustration de l’algorithme Min-Max.

coordonnées  $(x_i, y_i)$  et de la distance estimée  $d_i$  :

$$[x_i - d_i, y_i - d_i] \times [x_i + d_i, y_i + d_i] \quad (2.40)$$

La méthode détermine d’abord le maximum de tous les minimums et le minimum des maximums :

$$[\max(x_i - d_i), \max(y_i - d_i)] \times [\min(x_i + d_i), \min(y_i + d_i)] \quad (2.41)$$

Puis la position d’un nœud est associée au centre du carré.

### 2.4.2 Les techniques de proximité

La méthode de proximité est utilisée pour déterminer la position d’un nœud principalement en se basant sur la présence dans une zone particulière ou en se basant sur un phénomène physique d’une portée limitée. La position est choisie comme celle du nœud de référence [Gentile *et al.*, 2012]. L’exactitude de ces méthodes est liée à la densité des nœuds de référence et à la portée du signal.

CoO (Cell of Origin) est une méthode simple de positionnement utilisée dans les applications avec une faible exigence concernant l’exactitude. C’est une méthode utilisée dans les réseaux cellulaires pour positionner un nœud mobile.

La détermination du centroïde implique la connaissance des positions de plusieurs nœuds de référence dans la portée et le nœud est simplement positionné par la détermination du centroïde des positions. Plusieurs variantes existent en fonction de la façon dont le centroïde est calculé [He *et al.*, 2003, Niculescu et Nath, 2003].

Les terminaux mobiles intelligents sont munis de caméras permettant d'exploiter les QRCODE, de transcepteurs RFID/NFC pouvant contribuer à positionner leurs porteurs par proximité ou à améliorer le positionnement.

### 2.4.3 Les méthodes de fingerprinting

Les techniques, dites du *fingerprinting*, sont des algorithmes de reconnaissance des formes permettant à partir des vecteurs obtenus par les techniques de profilage (voir section 2.3.5), de déterminer la position d'une entité.

Elles se composent de deux phases :

1. Une phase de calibration consistant en une campagne de mesures de la puissance et d'autres caractéristiques du signal. Elle est effectuée afin de déterminer une empreinte identifiant l'environnement. La campagne est menée soit par déplacement d'opérateurs humains dans l'environnement ou automatiquement par des robots.
2. La détermination de la position consiste à utiliser un classifieur pour déterminer le vecteur (associé à une position) appartenant à l'empreinte qui soit le plus proche du vecteur fourni par l'entité. Plusieurs algorithmes [Brunato et Battiti, 2005] ont été utilisés, par exemple un classifieur KNN par [Bahl et Padmanabhan, 2000].

Un choix simple pour l'estimation de la position consiste à prendre la position associée au vecteur de la base de données qui minimise la distance euclidienne avec le vecteur mesuré.

Les méthodes de fingerprinting ont plusieurs avantages par rapport aux autres méthodes : robustesse face aux NLOS, positionnement plus exact, facilité d'implémentation en utilisant le WiFi. Au fil du temps, l'apprentissage permet d'améliorer les résultats du système. Néanmoins, elles ont certains inconvénients :

1. la phase de calibration consomme du temps ;
2. l'estimation de la position ne peut être effectuée que dans les zones pour lesquelles des données pré-enregistrées existent ;
3. finalement, si des modifications surviennent dans l'environnement, leur exactitude est fortement diminuée. Il faudra aussi refaire le *fingerprint*.

L'approche dite « *crowd-source* » ou « organique » [Ledlie *et al.*, 2012], ou « folksonomique » ou encore « *Wikipedia-style* » [Bolliger, 2008] propose une solution permettant de réduire le temps de calibration et aussi la sensibilité aux modifications de l'environnement. Ainsi, au fur et à mesure du temps, la collaboration entre utilisateurs est sollicitée pour améliorer le système et permettre une constitution rapide de l'empreinte. Ces méthodes peuvent aussi être combinées avec d'autres, par exemple avec une méthode de proximité dans les zones pour lesquelles une empreinte n'est pas disponible.

Nous allons maintenant détailler quelque méthodes de *fingerprinting*.

### 2.4.3.1 Les méthodes probabilistes

Les méthodes probabilistes sont basées sur des probabilités conditionnelles pour résoudre le problème de classification. Nous avons vu qu'au moment de l'établissement du fingerprint, la puissance reçue peut être modélisée par une variable aléatoire suivant une loi de probabilité.

Soient :

- $x_i$  : la position courante du nœud. Cette position lui est inconnue et il y a  $n$  positions susceptibles d'être occupées par le nœud ;
  - $p(x_i)$  est la probabilité que le nœud soit à la position  $x_i$  ;
  - $\mathbf{rss}_i$  est le vecteur de signaux observés durant la phase de mesure ;
  - $p(x_i|\mathbf{rss}_i)$  : est la probabilité que le nœud soit à la position  $x_i$  sachant la valeur du vecteur  $\mathbf{rss}_i$ .
  - $p(\mathbf{rss}_i|x_i)$  : est la probabilité que la mesure  $\mathbf{rss}_i$  ait été effectuée à la position  $x_i$  ;
- Une première méthode pour déterminer la position est donnée par la règle suivante :

$$x_i \text{ tel que } i = \arg \max_{j=1 \dots n} p(x_j|\mathbf{rss}_i) \quad (2.42)$$

Une autre approche est d'utiliser la vraisemblance pour déterminer la position. La règle est donnée par :

$$x_i \text{ tel que } i = \arg \max_{j=1 \dots n} p(\mathbf{rss}_i|x_j) \quad (2.43)$$

En se basant sur cette vraisemblance, un estimateur de maximum de vraisemblance peut être déterminé de la façon suivante. En considérant que les unités de mesures sont indépendantes (Hypothèse qui n'est pas toujours vérifiée dû à la corrélation des effets de masques : voir chapitre 1 ) :

$$V = \prod_{i=1}^n p(rssi|x_i) \quad (2.44)$$

La probabilité conditionnelle  $p(rssi|x_i)$  est supposée connue et peut être déterminée de plusieurs façons. Soit directement, à partir des données, en utilisant la méthode par histogramme, soit en utilisant une distribution de probabilité théorique dont les paramètres sont déterminés à partir des données. C'est le kernel gaussien [Evennou, 2007] qui est le plus utilisé :

$$p(rssi|x_i) = \frac{1}{n} \sum K(rss, rssi) \quad (2.45)$$

$$K(rss, rssi) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(rss - rssi)^2}{2\sigma^2}\right) \quad (2.46)$$

La loi de Weibull est aussi utilisée pour caractériser la propagation des signaux.

### 2.4.3.2 La méthode des $k$ plus proches voisins

C'est la méthode introduite par Bahl dans [Bahl et Padmanabhan, 2000].

Soient :

- $\mathbf{rss}_m$  : le vecteur mesuré ;
- $\mathbf{rssi}$  : le vecteur associé à la position  $x_i$  ;
- $d(.,.)$  : est une distance euclidienne avec  $d(\mathbf{rssi}, \mathbf{rss}_m) = \sqrt{\sum_{j=1}^n (\mathbf{rssi}^j - \mathbf{rss}_m^j)^2}$

Pour prendre en compte l'absence de mesures dans la base de données pour certains nœuds de référence, une pénalité ( $\mathbf{rss}_p$ ) est introduite dans le calcul de la distance. Soit  $d_j$ , alors on a :

$$d_j = \begin{cases} 0 & \text{Si } \mathbf{rss}_m^j \text{ n'existe pas mais que} \\ & \mathbf{rssi}^j \text{ est disponible dans la} \\ & \text{base de données.} \\ \mathbf{rss}_m^j - \mathbf{rss}_p & \text{Si } \mathbf{rssi}^j \text{ n'est pas disp. dans} \\ & \text{la base de données pour} \\ \mathbf{rssi}^j - \mathbf{rss}_m^j & \text{Sinon} \end{cases}$$

La fonction :

$$d(\mathbf{rssi}, \mathbf{rss}_m) = \sqrt{\sum_{j=1}^M d_j^2}$$

Les  $k$  plus proches voisins sont déterminés par :

$$V_{m+1} = \left\{ \underset{x_i}{\operatorname{argmin}} [d(\mathbf{rssi}, \mathbf{rss}_m)], x_i \notin V_m \text{ et } m+1 \leq k \right\} \quad (2.47)$$

La position est déterminée par :

$$x = \frac{\sum_{j=1}^k \frac{1}{d(\mathbf{rssi}, \mathbf{rss}_m)} \times x_j}{\sum_{j=1}^k \frac{1}{d(\mathbf{rssi}, \mathbf{rss}_m)}} \quad (2.48)$$

Les techniques de *fingerprinting* sont nombreuses. Nous n'avons pas cité ici les classifieurs SVM, les réseaux de neurones, etc.

## 2.4.4 Les méthodes bayésiennes

### 2.4.4.1 Description générale du modèle

Les filtres bayésiens estiment, de façon probabiliste, l'état d'un système dynamique à partir d'un ensemble d'observations contenant des erreurs. L'incertitude de mesure est représentée par une distribution de probabilité définie sur l'état et appelée fonction *Belief* [Fox *et al.*, 2003].

L'état est représenté à un instant  $t$  par une variable aléatoire  $x_t$  et est associé à une fonction *belief*  $Bel(x_t)$  définie par la densité postérieure sur la variable aléatoire  $x_t$  et conditionnée sur toutes les données disponibles à l'instant  $t$  :

$$Bel(x_t) = p(x_t | z_{1:t}) \quad (2.49)$$

Où  $z_{1:t}$  représente les données sur l'état du système à l'instant  $t$ .

Pour l'estimation de la position, une approche simple consiste à considérer les états du système comme les positions d'une entité, et les observations sont fournies par les nœuds déployés dans l'environnement ou portés par l'entité. L'idée clé du positionnement par filtres bayésiens est d'estimer séquentiellement la position en se basant sur la fonction *Belief* et l'espace de l'état conditionné par les informations contenues dans les données obtenues à partir des nœuds. Ainsi, les effets multi-trajets (NLOS) sont pris en compte par la méthode.

Ainsi, la fonction *Belief* apporte la réponse à la question : « Quelle est la probabilité qu'une entité soit à la position  $x_t$ , si l'historique des observations à partir des nœuds est  $z_{1:t}$  pour toutes les positions possibles pour  $x$  ? » [Fox *et al.*, 2003]

En général, la complexité du calcul de telles densités postérieures grandit exponentiellement en fonction du temps car le nombre de mesures à partir des nœuds croît aussi avec le temps. Pour rendre les calculs résolubles, les filtres bayésiens supposent que le système dynamique est Markovien<sup>3</sup>.

La mise à jour des filtres se fait en deux étapes :

1. la prédiction : à chaque instant de la mise à jour, l'état est « prédit » suivant la règle suivante :

$$Bel^-(x_t) \rightarrow \int p(x_t|x_{t-1})Bel(x_{t-1})dx_{t-1}$$

Ici, le terme  $p(x_t|x_{t-1})$  décrit les dynamiques du système, à savoir comment le système change au cours du temps. En estimation de la position, la probabilité conditionnelle est le modèle de déplacement (*motion model*), indiquant où la personne devrait être à l'instant  $t$  sachant qu'elle était à la position  $x_{t-1}$  précédemment. Le modèle de déplacement dépend fortement des informations disponibles pour le processus d'estimation. La prédiction peut aller de la prédiction de la position suivante de la personne en utilisant les estimations de la vitesse de la personne à la prédiction du moment où la personne va quitter par exemple le bâtiment en utilisant l'estimation du *but* de la personne.

2. la correction : Chaque fois que de nouvelles informations  $z_t$  sont disponibles en provenance des nœuds, les mesures sont utilisées pour prédire le *belief* :

$$Bel(x_t) = \alpha_t p(z_t|x_t)Bel^-$$

$p(z_t|x_t)$  est le modèle perceptuel qui décrit la vraisemblance de faire l'observation  $z_t$  sachant que la personne est à la position  $x_t$ . Pour l'estimation de la position, le modèle perceptuel est considéré comme une propriété de la technologie des nœuds utilisés. Il dépend des types et des positions des nœuds et capture les caractéristiques des erreurs dues à ce type spécifique de nœuds. Le terme  $\alpha_t$  est simplement un terme de normalisation qui assure que la somme des probabilités soit égale à 1.

$Bel(x_0)$  est initialisé en ayant une connaissance *a priori* de la position de la personne. Si aucune information n'est disponible, la position est considérée uniformément distribuée. Les filtres bayésiens sont des concepts abstraits dans le sens où ils ne fournissent qu'une plateforme probabiliste pour l'estimation récursive d'un état.

---

3. L'hypothèse de Markov consiste à supposer que toutes les informations pertinentes sont contenues dans l'état courant disponible  $x_t$

Les paramètres des modèles perceptuels et de déplacement peuvent être appris à partir des données en utilisant la méthode EM (Expectation-Maximization), une méthode populaire d'estimation à partir de données incomplètes. Le modèle perceptuel  $p(z_t|x_t)$  est généralement indépendant de la personne et peut être appris avant (*beforehand*). Le modèle de déplacement, de l'autre côté, peut être différent pour chaque personne. L'apprentissage des paramètres du modèle de déplacement permet au système de s'adapter à une personne spécifique, ceci en augmentant l'exactitude et l'efficacité du processus d'estimation [Fox *et al.*, 2003].

Pour implémenter ces filtres, les trois modèles doivent être spécifiés, à savoir : le modèle perceptuel  $p(z_t|x_t)$ , les dynamiques du système  $p(x_t|x_{t-1})$  et la représentation de la fonction belief  $Bel(x_t)$ . Les propriétés des différentes implémentations des filtres bayésiens diffèrent fortement selon la façon dont ils représentent les densités de probabilités sur l'état  $x_t$ .

### 2.4.4.2 Les variantes des filtres bayésiens

Dans cette section, nous allons donner un bref aperçu des représentations de fonctions *beliefs* par les diverses implémentations des filtres bayésiens.

Les *filtres de Kalman* (KF) sont les variantes des filtres bayésiens les plus largement utilisés [Fox *et al.*, 2003, Evennou, 2007]. Ils approximent la fonction *belief* par des distributions gaussiennes unimodales, représentées par leur moyenne et variance. La moyenne donne la position attendue de la personne, la variance représente l'incertitude dans l'estimation de la position. Même si les filtres de Kalman font des hypothèses fortes sur la nature des nœuds et des déplacements de la personne, ils ont été appliqués avec grand succès à divers problèmes d'estimation [Fox *et al.*, 2003]. L'avantage principal des filtres de Kalman est leur efficacité en terme de calcul, qui est obtenue cependant aux dépens d'une puissance de représentation réduite car les KF ne peuvent représenter que des distributions unimodales. Il ne reste pas moins que les KF sont meilleurs dans les cas où l'incertitude dans la position de la personne n'est pas trop élevée.

Les filtres *Multi-hypothesis Tracking* (MHT) étendent les filtres de Kalman pour utiliser des fonctions *belief* à distribution multimodales [Macagnano *et al.*, 2012]. MHT représente le belief par des mélanges de gaussiennes où chaque hypothèse est traquée en utilisant un filtre de Kalman. Les poids des hypothèses sont déterminés par le degré de prédiction des mesures des nœuds. Grâce à leur capacité à représenter des *belief* multimodales, les approches MHT sont plus largement applicables que les KF [Fox *et al.*, 2003].

Les filtres *Grid-based* dépassent les restrictions imposées par les KF, en se basant sur des représentations discrètes et par pièces de la fonction *belief* [Morelli *et al.*, 2006]. Pour la localisation *indoor*, les méthodes basées sur des grilles font un découpage de l'environnement en petits espaces, couramment de tailles de 10cm à 1m [Fox *et al.*, 2003]. Chaque cellule de la grille contient la valeur du *belief* lorsque le nœud est dans la cellule. Un avantage clé de ces approches est qu'elles peuvent représenter des distributions arbitraires dans l'espace discret de l'état (chaque cellule pourrait avoir sa propre belief). Le désavantage de ces approches est la complexité de calcul qui ne les rendent applicables que dans les problèmes d'estimation à faible dimension comme l'estimation de la position ou de l'orientation. Ce qui est suffisant dans notre cas.

Les *particles filters* (PF) représentent les *beliefs* par des ensembles de distributions pondérées selon le *belief* [Madigan *et al.*, 2005]. Les PF réalisent les mises à jours des filtres de Bayes suivant une procédure d'échantillonnage. L'avantage clé des PF est leur capacité à représenter des densités de probabilité arbitraires qui les rendent applicables à des problèmes sur lesquels les KF ne sont pas bien adaptés. Comparés aux approches basées sur une grille, les PF sont très efficaces tant elles concentrent automatiquement leurs ressources sur les régions dans l'espace de l'état qui ont les fortes probabilités. Cependant, puisque les complexités des cas les plus mauvais de ces méthodes grandissent exponentiellement selon la dimension de l'espace d'état, il faut être précautionneux quant à l'application de ces méthodes sur des problèmes de grande dimension [Fox *et al.*, 2003].

## 2.5 Conclusion

Nous avons présenté, dans ce chapitre, les techniques de mesures couramment utilisées, et aussi un certain nombre d'algorithmes de positionnement, allant de méthodes simples à des algorithmes de reconnaissance des formes.

Comme nous l'avons déjà vu, le problème de la localisation peut se transformer en problème d'optimisation. Et, de ce fait, les méthodes bio-inspirées ont été appliquées, par exemple des algorithmes Particle Swarm Optimization (PSO) [Collette et Siarry, 2003, Kulkarni et Kumar, 2011, Garcia *et al.*, 2009], ACO [Mohanty et Parhi, 2013] et aussi des algorithmes génétiques [Nagy et Farkas, 2000, Fonlupt *et al.*, 1993].

En ce qui concerne la phase de raffinement de la position, elle est optionnelle et incluse dans certaines méthodes. Dans le chapitre suivant nous allons présenter un certain nombre de systèmes ayant mis en œuvre les méthodes que nous avons présentées et aussi nous allons donner des principes pour la conception d'un système de positionnement et son évaluation.

## Deuxième partie

# Mise en oeuvre de la localisation de personnes handicapées



## Chapitre 3

# Étude des Systèmes Ambient Assisted Living

### 3.1 Introduction

Ce chapitre est consacré à la présentation de l'*Ambient Assisted Living* ainsi qu'un certain nombre de systèmes AAL. Nous allons aussi voir quelques principes de conception de systèmes de positionnement.

### 3.2 La localisation des personnes à mobilité réduite

Nous rappelons, ici, que la localisation consiste à déterminer la position d'une entité connaissant les *estimations* de ses relations avec certains points de référence ou d'autres entités. Les entités que nous avons considérées, par la suite, sont des personnes à mobilité réduite.

La mobilité réduite est une situation temporaire ou permanente de handicap durant laquelle l'entité ne peut plus se servir de ses facultés motrices, sensorielles ou cognitives pour se déplacer d'un point à un autre.

Plusieurs travaux ont été entrepris pour offrir une accessibilité des nouvelles technologies aux personnes à mobilité réduite. Ce domaine d'application est connu sous le nom de *Ambient Assisted Living (AAL)*.

AAL est constitué de concepts, produits et services interopérables et combine les nouvelles technologies de l'information et de la communication et les environnements sociaux afin d'améliorer la qualité de vie des personnes vivant dans ces environnements [ERCIM, 2011]. Le concept d'AAL est à l'intersection de plusieurs domaines :

- domotique ;
- technologies de soins aux personnes à mobilité réduite ;
- les travaux sur l'accessibilité, l'utilisabilité et l'acceptabilité des nouvelles technologies ;
- l'intelligence ambiante qui offre de nouvelles possibilités pour fournir une assistance intelligente, non intrusive et ubiquitaire aux personnes à mobilité réduite et toute

entité en général.

Dans le cadre de cette thèse, nous nous sommes intéressés au positionnement de personnes à mobilité réduite, qui est un cas particulier du positionnement des entités. La particularité est la prise en compte des contraintes liées aux normes de sécurité et de sûreté de ses personnes pour chaque domaine d'application. Dans tous les cas, il faut d'abord déterminer une position de l'entité.

Les contraintes diffèrent en fonction du lieu et nous pouvons les regrouper en :

- espace de vie : maison, maison de retraite ;
- espace de soin : hôpital, maison de retraite médicalisée ;
- espace de travail : bureau, travail au domicile, etc.
- espace de loisir ;
- espaces publics : administration, etc

Bien que les contraintes soient différentes, le but est le même : offrir une aide afin de contrebalancer le handicap et ainsi permettre à la personne handicapées d'atteindre son objectif. Ceci reste pour le moment difficile à réaliser car les technologies de positionnement *indoor* ont des résultats différents, des coûts parfois très élevés, et ne prennent pas en compte les particularités des personnes handicapées. Une autre difficulté est le fait que les performances ne sont pas garanties lorsqu'on change d'un espace à un autre. Comment compenser le manque d'un système de positionnement dans un milieu donné ? Est-ce que tous les milieux ont la même technologie ? Plusieurs questions de la sorte se posent.

Dans la suite, nous allons présenter quelques systèmes ayant pour but de faciliter la vie des personnes handicapées.

### 3.3 État de l'art de systèmes de AAL

A. Marco *et al* ont présenté dans [Marco *et al.*, 2008], un système de localisation interne pour personnes âgées et handicapées appelé Zigbee and UltraSound Positioning System (ZUPS). ZUPS fournit : i) un système d'alarme pour surveiller la résidence et le comportement des résidents dans le but de lancer des alarmes lorsque des comportements à risque ou inappropriés se produisent et sont détectés par les capteurs ; ii) un système de guidage vers les sorties d'urgence en cas de problème, ou pour visiter la résidence, etc. iii) un système de détente permettant aux résidents de faire des exercices physiques et des compétitions ou des jeux de société.

Ambient Support Living System (ALS) implémente le concept de l'Intelligence Ambiante (Ambiant Intelligent) qui a pour but de développer un environnement pour personnes âgées et handicapées. L'architecture AmbienNet [Abascal *et al.*, 2009] qui supporte ALS est basée sur : i) deux réseaux de capteurs qui fournissent les informations sur les paramètres physiques (température, humidité, etc.) de l'environnement, un système de localisation interne et de navigation ; ii) un middleware pour intégrer les divers matériels hétérogènes ; iii) des fauteuils roulants intelligents ayant des sonars et des capteurs infrarouges.

**SmartAssist** [Schrader *et al.*, 2010] est un projet qui a été réalisé à l'université de Luebeck en Allemagne. Ses motivations ont été de prendre en compte les aspects positifs et

négatifs de l'accroissement de la durée de vie. Les aspects négatifs se traduisent par la surcharge des services de soins qui n'arrivent pas à répondre correctement aux attentes de toutes les personnes âgées, la diminution de l'autonomie dans certains cas. Tandis que les aspects positifs sont, entre autre, le temps qui peut être consacré à des activités privées, professionnelles et sociales. Pour ce faire, le système se base sur :

- **Un réseau de capteurs** non intrusif, c'est-à-dire n'utilisant pas de capteurs caméra et microphone. Des capteurs de petite taille sont déployés dans la maison et mesurent la température, la pression, la proximité, la luminosité, l'humidité et l'électricité. Les smartphones et autres terminaux Android peuvent aussi être intégrés au réseau en tant que capteurs en utilisant Dynamix [Carlson, 2013]. Ces capteurs sont capables de créer automatiquement le réseau de façon collaborative. Une station de base permet de collecter les données et de les envoyer au serveur de SmartAssist. Les données ainsi récoltées sont traitées afin de déterminer des événements (machine à café mise en marche, porte ouverte, sonnerie à la porte, etc) et de lancer les actions appropriées appeler le médecin, le voisin, etc. L'ensemble des événements déterminés définit le contexte de l'utilisateur. Des patrons sont enregistrés dans le système et contiennent les descriptions des caractéristiques des utilisateurs. Un patron peut changer dynamiquement d'état pour être un producteur ou un consommateur de services. Par exemple, le voisin peut se proposer pour aider la personne âgée dans le cas où son profil a été enregistré et les accès accordés. Et la personne âgée en retour se propose pour donner des cours de soutien scolaire à son enfant.
- Un portail Web servant d'environnement numérique pour les utilisateurs, d'un outil de communication entre membres du réseau social et d'un point d'intégration pour les développeurs. Ce portail est basé sur Druppal 6 [Druppal Fondation, 2013] et du conteneur Apache Shindig [Appache Foundation, 2013] qui est une implémentation libre du standard OpenSocial [Opensoicial Foundation, 2013]. Cette dernière permet le développement de réseaux sociaux ouverts et interopérables en fournissant une spécification XML, une API Javascript permettant de développer des Gadgets pour l'interaction entre les utilisateurs, le partage des données sur les utilisateurs et les capteurs. La préservation de la vie privée est assurée par OAuth [OAuth, 2013].
- Un middleware nommé Aladdin « sensible au contexte » et qui permet le développement de plusieurs applications. La plateforme est illustrée par la Figure 3.1 : Il est installé sur chaque appareil et, à l'exécution, il détermine les caractéristiques de celui-ci et installe les *plugins* nécessaires aux tâches demandées à l'appareil en fonction de ses capacités. Aladdin analyse continuellement l'environnement de l'utilisateur afin de déterminer le contexte de l'utilisateur et de le fournir à l'application supérieure, ce qui permet à celle-ci d'adapter son comportement à ce contexte.

SmartAssist a été exploité entre autres pour la surveillance de la santé des personnes âgées et aussi pour une application de jeu sportif nommé BikeWars qui compare les résultats cardiaques des compétiteurs.

**SM4All** (Smart Home for All) [SM4ALL Project, 2013] est un projet visant à faciliter la mise en place de solution AAL dans les maisons. En effet, sur la base du constat que plusieurs types de systèmes existent, aussi bien open source que propriétaire, que plusieurs fabricants ferment leurs appareils, une architecture à middleware a été proposée. Cette ar-

### 3.3. ÉTAT DE L'ART DE SYSTÈMES DE AAL

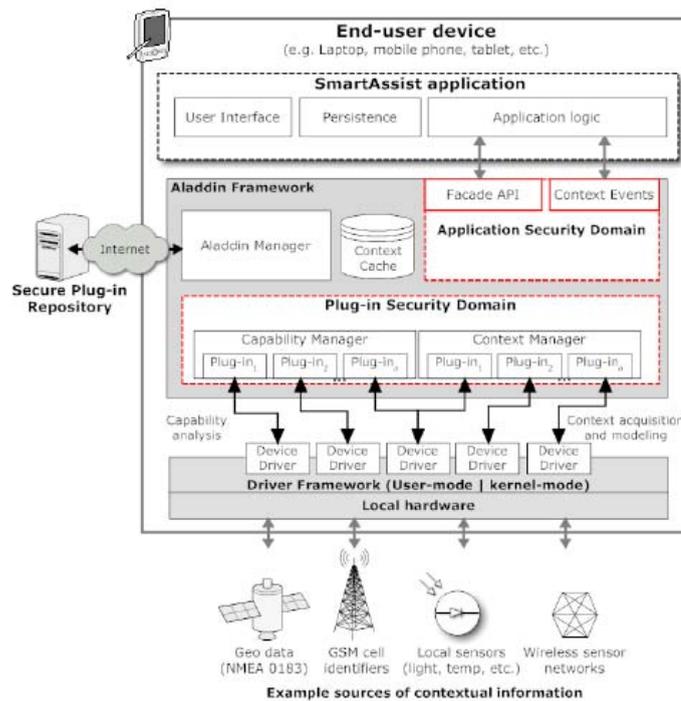


FIGURE 3.1 – La plateforme Aladdin. Source [Schrader *et al.*, 2010]

architecture a pour ambition de développer et de structurer l'intelligence ambiante dans un environnement. L'idée est de faire travailler ensemble tous les appareils disponibles dans une maison et de coordonner leur fonctionnement de façon automatique afin d'exécuter des tâches complexes qui font appel à d'autres tâches plus simples. Par exemple, comme scénario, une personne handicapée peut communiquer avec sa maison, munie de SM4All, afin de se faire préparer son bain (nécessite de faire couler l'eau, de la chauffer à la température désirée, etc) et de se faire aider si besoin.

La figure 3.2 illustre l'architecture de SM4All.

La couche interface utilisateur est composée de plusieurs types d'interfaces comme des actionneurs, des terminaux tactiles et aussi des *Brain Computer Interfaces* (BCI). Les BCI permettent une interaction entre le cerveau et le système et sont d'une grande utilité pour les personnes handicapées. Un des objectif du projet est l'évolutivité afin de permettre l'intégration d'autres interfaces et appareils.

La couche de composition est chargée de déterminer les autres sous-tâches dont une tâche a besoin.

La couche *pervasive* utilise une abstraction du matériel à base de service SOAP, UPnP et WSDL. Aussi bien le matériel, les utilisateurs que les logiciels peuvent fournir des services. Le proxy joue l'interface entre le demandeur et le fournisseur de service afin de leur fournir dans le premier cas le service et dans le second l'interface appropriée.

### 3.4. PRINCIPES DE CONCEPTION D'UN SYSTÈME DE LOCALISATION

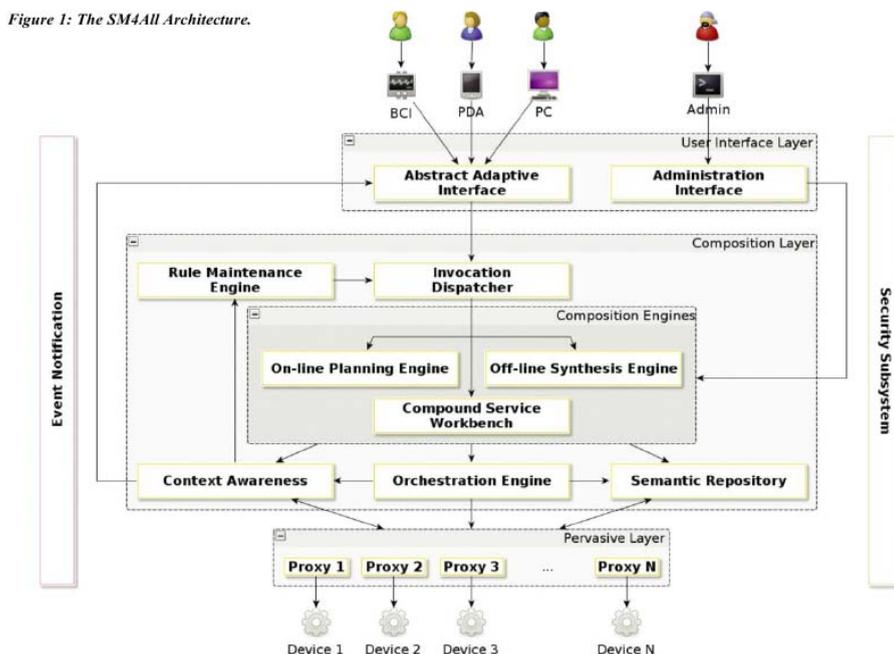


FIGURE 3.2 – L’architecture de SM4All. Source [ERCIM, 2011]

**Intelligence Bed** est une proposition de système de surveillance qui est basée sur la capture et l’interprétation de la distribution de la charge exercée sur les quatre pieds du lit. Le lit est équipé de quatre capteurs iLoad du fabricant Loadster Sensors. Les quatre capteurs sont connectés à une station centrale chargée de lancer les mesures et d’autres fonctionnalités des capteurs.

Le système peut être utilisé pour déterminer :

- si le lit est occupé ou non. Ceci par la mesure de la charge perpendiculaire excédant celle créée par le lit ;
- à quels moments le lit est occupé ou non ;
- comment la personne se déplace dans le lit. Ceci par mesure de la déflexion perpendiculaire sur la zone de distribution de la charge.

Les données ainsi collectées servent à créer des modèles, par exemple, les données décrivant comment la personne tourne dans son lit, se lève, etc. Des algorithmes de reconnaissance des formes peuvent être utilisés pour déterminer le cas.

La bonne conception du middleware est importante pour les systèmes basés sur la position. Dans [Pfeifer, 2005], l’auteur présente une architecture redondante utilisant un middleware pour intégrer plusieurs technologies : GPS, RFID, caméra, réseaux de capteurs, etc. afin d’obtenir de meilleurs résultats.

### 3.4 Principes de conception d’un système de localisation

La conception d’un système de localisation *indoor* doit prendre en compte les besoins des utilisateurs ainsi que les exigences pour chaque domaine d’application.

Mautz [Mautz, 2009] a établi 16 critères pour évaluer et comparer différents systèmes de localisation à l'intérieur de bâtiment. Ce grand nombre de critères rend difficile pour un utilisateur l'identification et le choix du système optimal pour une application particulière. Ces 16 critères synthétisent plusieurs travaux entrepris par des auteurs différents et à des étapes différentes de l'évolution de la recherche sur les systèmes de positionnement interne dans le souci d'évaluer proprement ces systèmes [Hightower et Borriello, 2001, Pahlavan *et al.*, 2002, Bensky, 2007, Liu *et al.*, 2007, Mao *et al.*, 2007, Valaee et Li, 2012].

Dans [Mautz, 2012], les auteurs précisent que pour trouver une technologie de positionnement appropriée pour une application particulière, les paramètres de performance doivent être mis en adéquation avec les besoins de l'utilisateur. Ces paramètres posent une optimisation multidimensionnelle lorsque que l'on cherche la meilleure adéquation. Par ailleurs, les valeurs pour les paramètres de performance ne sont pas toujours exactement déterminables car ils dépendent de plusieurs facteurs, circonstances et conditions. Chaque approche du système n'a pas seulement son ensemble de paramètres individuels mais aussi des caractéristiques uniques, conditions, hypothèses et applications qui doivent être pesées les unes contre les autres. La pondération de tous les paramètres et des conditions additionnelles ne peut pas être faite de façon objective. De ce fait, un classement impartial (*fair-minded*) n'est ni utile ni faisable. Chacun devra adapter son système à ses utilisateurs.

#### 3.4.1 Exactitude (*accuracy*) / Précision (*precision*)

Un des critères importants pour l'évaluation de la qualité d'un positionnement est la mesure de la proximité de la position donnée avec la vraie position. Deux notions sont principalement utilisées pour évaluer cette étroitesse et sont souvent confondues dans le domaine du positionnement interne. Ce sont l'exactitude et la précision. Une description plus étendue de l'utilisation de ces notions dans ce domaine est donnée dans [Chen *et al.*, 2003].

*L'exactitude*, comme définie par le Groupe de travail 2 du comité commun de guides en métrologie (JCGM/WG 2) [JCGM, 2008], est l'étroitesse entre une valeur mesurée et une valeur vraie d'un mesurande. Cependant, l'exactitude n'est pas une grandeur et ne s'exprime pas numériquement. Un mesurage est quelquefois dit plus exact s'il fournit une plus petite erreur de mesure. Celle-ci est la différence entre la valeur mesurée d'une grandeur et une valeur de référence.

*La précision* est quant à elle définie comme l'étroitesse de l'accord entre les indications ou les valeurs mesurées obtenues par des mesurages répétés du même objet ou d'objets similaires dans des conditions spécifiées. La précision est en général exprimée numériquement par des caractéristiques telles que l'écart-type, la variance ou le coefficient de variation dans les conditions spécifiées. Les conditions spécifiées peuvent être, par exemple, des conditions de répétabilité, des conditions de fidélité intermédiaire ou des conditions de reproductibilité. La précision sert à définir la répétabilité de mesure, la fidélité de mesure et la reproductibilité de mesure.

Pour Mautz [Mautz, 2012], le terme exactitude de positionnement a été utilisé pour exprimer le degré de conformité à la vraie valeur de la valeur d'une position estimée ou mesurée à un instant donné. Il est exprimé pour les composantes verticales et horizontales au seuil de confiance de 95%. Si nous considérons une distribution normale, la qualité de

### 3.4. PRINCIPES DE CONCEPTION D'UN SYSTÈME DE LOCALISATION

Déploiement	Distance entre nœuds	Couverture de chaque nœud
Rectangulaire	$r$	$r^2$
Triangulaire	$r$	$\frac{\sqrt{3}}{4}r^2$
Hexagonale	$r$	$\frac{\sqrt{3}}{3}r^2$

TABLE 3.1 – Les divers types de déploiement réguliers avec les couvertures pour chaque nœud.

la position est exprimée par :

$$\sigma_p = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{x}_i - x_i)^2}, \quad (3.1)$$

où  $n$  est le nombre de positions estimées (i.e mesurées),  $\hat{x}_i$  et  $x_i$  le vecteur de position prédit par le modèle pour le nœud localisé  $i$ , ou si une seule position  $x_i$  est estimée,  $x_i$  est remplacé par un simple vecteur de position  $x_0$ . Un critère qui est moins sensible aux valeurs aberrantes est la moyenne des écarts absolus des positions :

$$a_p = \frac{1}{n} \sum_{i=1}^n |\hat{x}_i - x_i|, \quad (3.2)$$

De ce fait, une faible exactitude fait référence à  $\sigma_p > 10$  m et une grande précision à  $\sigma_p < 1$  cm. Cependant, pour plusieurs applications, une précision de niveau chambre est suffisante.

#### 3.4.2 La couverture (*coverage*)

En fonction de l'application, la couverture dépend de certaines caractéristiques des nœuds, de l'environnement et du type de déploiement choisi.

La couverture décrit l'extension spatiale où les performances du système doivent être garanties. La couverture se décline en trois catégories :

- Couverture locale : zone de déploiement petite et bien délimitée qui n'est pas extensible (chambre ou immeuble). Pour ce cas, la taille de la couverture est spécifiée par exemple en mètres,  $m^2$  ou  $m^3$ . L'extensibilité est liée à des considérations surtout économiques puis de disponibilité d'espace suffisant pour déployer des nœuds supplémentaires et des interférences occasionnées.
- Couverture extensible : possibilité d'augmenter la taille de la zone de déploiement en l'étendant par un déploiement de matériels supplémentaires. Pour ces deux premiers types de couverture, nous allons considérer un déploiement régulier. Bien qu'un déploiement aléatoire est envisageable en localisation interne, pour les raisons indiquées dans la section 3.4, nous n'allons pas l'utiliser en pratique. Un déploiement régulier suit trois schémas [Agrawal, 2011] : rectangulaire, triangulaire ou hexagonale et les couvertures sont indiquées dans le tableau 3.1. Étant entendu qu'en pratique, ces schémas sont à adapter au domaine de déploiement. Il en est

de même pour la distance  $r$  qui n'est pas toujours facile à déterminer puisque les nœuds sont hétérogènes.

- Couverture globale : le système opère à l'échelle globale comme les systèmes GNSS.

### 3.4.3 L'intégrité (*integrity*)

L'intégrité fait référence à la confiance qui peut être placée dans les sorties du système. Le risque lié à l'intégrité est la probabilité qu'une mauvaise localisation conduise à une position estimée qui diffère de la position voulue de plus d'une valeur acceptable (valeur d'alarme) et que l'utilisateur n'en ait pas été informé dans une période de temps (temps d'alarme). Dans le domaine du positionnement, il est difficile de trouver des valeurs quantifiées pour le critère d'intégrité.

### 3.4.4 La disponibilité (*availability*)

La disponibilité est le pourcentage de temps durant lequel le service de positionnement est disponible pour l'usage avec l'exactitude et l'intégrité requises. Elle peut être limitée par des facteurs aléatoires (les pannes, la congestion des communications) aussi bien que des facteurs programmés (les maintenances de routines). Généralement, un des trois niveaux suivants est spécifié, bien que cela dépende du domaine d'application :

- Faible  $< 95\%$  ;
- Normal  $> 99\%$  ;
- Grande  $> 99.99\%$ .

Pour atteindre la disponibilité, il est supposé que la continuité, l'exactitude et l'intégrité sont satisfaites. Les descriptions des applications comportent des spécifications de la disponibilité, mais des figures des disponibilités ne sont pas toujours fournies.

### 3.4.5 La continuité

La continuité est la propriété d'un fonctionnement continu du système sur une période de temps pour exécuter une fonction spécifiée. La fréquence des coupures acceptables devrait être donnée. La continuité est similaire à la disponibilité.

### 3.4.6 Le taux de mise à jour (*Update Rate*)

Le taux de mise à jour est la fréquence à la quelle les positions sont déterminées sur le nœud ou par l'infrastructure et communiquées à l'entité. Les mises à jour sont effectuées :

- périodiquement : une mise à jour régulière, spécifiée dans une durée (par exemple en secondes) ;
- à la demande : déclenchées par l'utilisateur ou par un matériel lointain ;
- suite à un événement : la mise à jour est effectuée lorsqu'un événement particulier survient par exemple lorsque la valeur de la température excède une valeur seuil.

### 3.4.7 Le temps de latence du système

Le temps de latence du système décrit le délai avec lequel les informations demandées par l'utilisateur lui sont fournies. Le temps de latence peut être de l'un des types suivants :

- temps réel : ne tolère pas les délais "perceptibles". C'est le type de temps de latence le plus exigeant. Il est nécessaire pour la plupart des applications de positionnement interne et de la navigation.
- au plus tôt (*Sooner the better*) : requiert les meilleurs efforts du système ;
- au plus tôt mais avec une limite supérieure : demande les meilleures efforts au système et celui-ci doit être conçu pour limiter le délai maximum à un seuil ;
- libre (Post processing) : aucun temps pour la réponse n'est spécifié.

### 3.4.8 Les données de sortie (*Output Data*)

Un système de positionnement peut en plus de la position fournir d'autres informations rentrant dans le cadre des LBS. Ainsi, la vitesse (*speed/velocity*), l'accélération, la direction (*Heading/bearing*) peuvent être fournies. Ces informations complémentaires sont très utiles pour certaines applications.

### 3.4.9 Les interfaces homme-machine (*Interface*)

Les interfaces Hommes-Machines décrivent comment les informations sur la position doivent être reportées et la forme des requêtes sur le nœud de l'entité. Ces questions serviront de guide pour la description d'une IHM :

- Un affichage graphique est-il requis ?
- Est-ce qu'une carte à l'échelle suffit ou une exactitude topologique est suffisante ?
- Est-ce que des informations cartographiques supplémentaires sont nécessaires ?
- Quel niveau de détail est requis ?
- Est-ce que des données sur les états du réseau et des terminaux seront requises ?
- Est-ce que les informations de planification des routes sont requises ?
- Est-ce que le guidage (visuel ou sonore) est requis ?
- Les instructions en langage naturel (*Natural Language Instructions* : NLI) sont-elles requises ? NLI est une façon commode de donner des informations aux utilisateurs, offrant ainsi un moyen riche et flexible pour décrire les routes de navigation.

### 3.4.10 Sécurité et Préservation de la vie privée

La sécurité d'un système est l'ampleur de la protection contre toute action de nature malveillante ayant pour but les invasions de la vie privée, le vol, et la corruption des informations ou des dommages physiques. Des mesures doivent être décrites pour assurer la sécurité d'un système de positionnement dans son domaine de déploiement. Par contre, la sûreté d'un système adresse les risques liés aux actions non malveillantes qui pourront survenir [Stallings et Brown, 2008].

La préservation de la vie privée est un problème clé pour les systèmes de positionnement. L'utilisateur doit être associé, de façon à ce qu'il soit au courant de la nature des

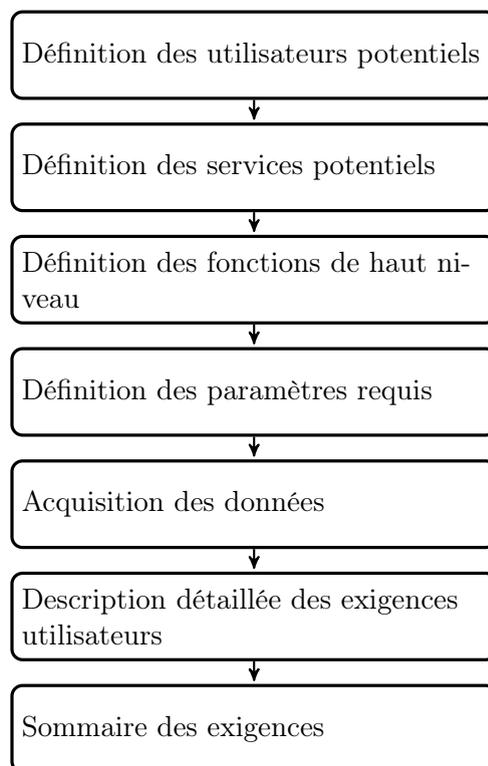


FIGURE 3.3 – Le processus de la définition de besoins des utilisateurs. D’après [Mautz, 2012]

informations sur sa vie privée qui sont requises par l’application, et ainsi pouvoir donner son approbation. Il faut aussi prendre en compte les particularités de certains domaines concernant les données des clients, des visiteurs, etc. [Beresford et Stajano, 2004]

#### 3.4.11 Les coûts

Le coût maximal d’un système de positionnement est un critère important qui peut être déterminé de plusieurs façons. Les coûts en temps incluent le temps d’installation et de maintenance. Les coûts en capitaux comprennent le coût unitaire par nœud, l’infrastructure et les salaires du personnel de support. Le coût de maintenance inclut les dépenses pour maintenir le système fonctionnel. Le coût en espace inclut le nombre d’infrastructures installées et la taille du matériel. L’estimation des coûts doit être menée soigneusement en prenant en compte des dépendances du temps et des positions aux contraintes des fabricants.

### 3.5 Processus de définition des besoins utilisateurs

La figure 3.3 montre une approche générale pour définir les exigences des utilisateurs. Premièrement, le groupe des utilisateurs potentiels est défini et listé. Basés sur les groupes

### 3.6. CONCLUSION

---

utilisateurs, les services associés à ces groupes sont déterminés. Ensuite les fonctions minimales qu'un système de positionnement interne doit remplir sont définies. À partir de ces fonctions, une liste de paramètres pour capturer les besoins utilisateurs sont définis. L'acquisition de données est menée à partir d'une combinaison de sources. Premièrement, une étude des utilisateurs est effectuée par des questionnaires, des interviews et des sessions de réflexion et des interviews avec des partenaires.

L'évaluation des questionnaires, des interviews et des sessions avec les groupes d'utilisateurs est menée pour chaque groupe d'utilisateur et chaque application séparément. Les résultats d'une telle étude constituent les paramètres des utilisateurs dans une forme explicite. Dans les sections suivantes nous allons décrire les composantes de bases d'un système de positionnement, comme illustré par la figure 2.4.

## 3.6 Conclusion

Ce chapitre a constitué une présentation des besoins des personnes handicapées ainsi qu'un certain nombre de travaux dans le domaine du AAL. Nous avons aussi présenté quelques principes de conception d'un système de positionnement. Le chapitre suivant est consacré à nos travaux.

### 3.6. CONCLUSION

---

## Chapitre 4

# AIPS : Vers Un Autre Système de Géolocalisation Interne et d'Assistance à la Mobilité

### 4.1 Introduction

Ce chapitre traite des travaux que nous avons entrepris afin de mettre en place un système de positionnement. Il montre l'évolution de nos travaux qui se sont d'abord intéressés aux réseaux de capteurs, au sens strict, puis une approche par middleware prenant en compte tout appareil capable d'émettre dans le bâtiment. Ces travaux ont été publiés dans [Amadou Kountché *et al.*, 2011].

### 4.2 Première proposition

Ce prototype est la mise en œuvre directe d'un certain nombre de méthodes que nous avons décrites dans le Chapitre 2.

Le prototype utilise un réseau de capteurs sans fil pour localiser un smartphone Android Nexus One. Les nœuds du réseau de capteurs sont munis de modules GSM/GPRS leur permettant d'être reliés au même réseau GSM que le smartphone. Un réseau GSM est un ensemble de cellules munies entre autre d'un identifiant et d'une station de base. Chaque appareil qui se trouve dans une cellule donnée peut être identifié par l'identifiant de la cellule. Nous allons utiliser cet identifiant qui est le même à un moment donné pour le smartphone et au moins trois nœuds de référence. Après avoir estimé la distance séparant les nœuds du smartphone à partir des fonctions fournies par l'API des capteurs, nous appliquerons les méthodes géométriques décrites dans le Chapitre 2. Nous avons aussi expérimenté avec ce prototype la méthode de raffinement de la position : Broyden-Fletcher-Goldfarb-Shanno (BFGS) qui est une méthode d'optimisation sans contrainte. L'algorithme de détermination des distances est décrit dans la sous-section consacrée au réseau de capteurs tandis que BFGS est décrite dans cette sous-section.

La zone de déploiement est le rez-de-chaussée du Département Informatique de l'École

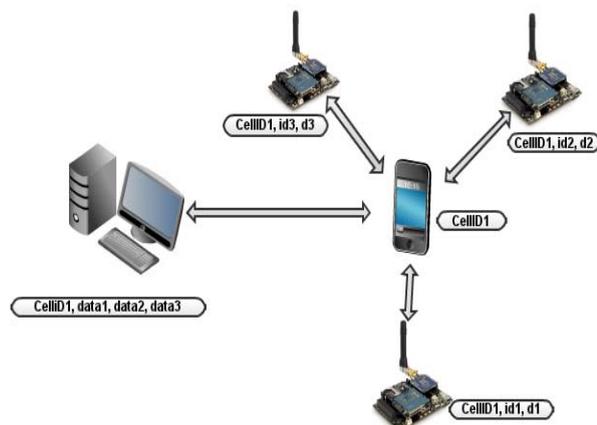


FIGURE 4.1 – Cette figure montre le processus de la localisation par identifiant de cellule GSM.



FIGURE 4.2 – L’architecture du prototype utilisant un réseau de capteurs.

Polytech’Tours. Le prototype est constitué de trois parties : le superviseur, le réseau de capteurs sans fil Zigbee et l’application. La Figure 4.2 décrit l’architecture choisie.

D’une part, le choix de la combinaison GSM/Zigbee est lié au fait que le smartphone ne dispose pas de *transceiver* Zigbee. Bien qu’il soit facile de mettre en œuvre un positionnement par réseau GSM en identifiant le numéro de la cellule<sup>1</sup>, et en associant la position de la cellule à celle du smartphone, c’est la première fois, à notre connaissance que cette approche est adoptée. D’autre part, c’est la technologie des réseaux de capteurs que nous avons voulu utiliser pour le positionnement *indoor*. C’est ainsi que nous avons fait l’acquisition de plusieurs capteurs Waspnote.

#### 4.2.1 L’application Superviseur

Le superviseur est responsable de la surveillance de l’état du système et des utilisateurs.

L’état du système est constitué de l’état des capteurs :

- fonctionne ou non ;
- le niveau d’énergie : alimenté par le secteur ou par batterie ;
- arborescence des nœuds ;

1. Des identifiants de cellules sont disponibles sur <http://www.opencellid.org>

— rôle : coordinateur, nœud de référence, nœud feuille ;

La surveillance du système consiste aussi à journaliser l'activité des nœuds (niveau d'énergie disponible, temps, etc), envoyer des requêtes aux capteurs, reprogrammer les nœuds si besoin par *Over the Air Programming* et les incidents survenus.

La surveillance des utilisateurs consiste à déterminer leurs positions et à leur envoyer des avertissements en cas de problèmes. Ces avertissements peuvent être de plusieurs types : des messages SMS, un appel sur le téléphone. Il faut noter que les capteurs du smartphone peuvent être exploités afin d'affiner cette surveillance. Les divers capteurs (notamment l'accéléromètre et le gyroscope) peuvent servir à déterminer l'état du déplacement de la personne : en mouvement, immobile, tombé.

Le superviseur reçoit les informations de localisation via une station de base (nœud du réseaux de capteurs) connectée à un port USB.

Les informations de positionnement reçues par le superviseur sont enregistrées dans un fichier suivant le formant :  $CellID_i, data_1, data_2, data_3$  où  $data_i, i = 1, 2, 3$  est une structure de données contenant la distance  $d_i$  et l'identifiant  $id_i$  du nœud  $i$ .

#### 4.2.1.1 La méthode BFGS

La position obtenue est raffinée en utilisant l'algorithme de Broyden-Fletcher-Goldfarb-Shanno (BFGS).

BFGS, comme Davidon-Fletcher-Powell, sont des méthodes Quasi-Newton elles-même sous-catégories des méthodes de Newton. Une méthode de Newton détermine itérativement la solution d'une équation  $f(x) = 0$  par :

$$x_{k+1} = x_k - \nabla f(x_k)^{-1} \times f(x_k) \quad (4.1)$$

$f(x)$  est définie de  $R^n \rightarrow R^n$ .

Les limitations de la méthode de Newton interviennent lorsque la dimension  $n$  augmente induisant un grand temps de calcul de la matrice jacobienne (la matrice des dérivées primaires de la fonction) ainsi que la difficulté du calcul du second terme de l'équation.

Les méthodes de Quasi-Newton remédient à cela en remplaçant  $\nabla f(x)$  par une matrice,  $M_k$ , qui est une estimation de l'inverse du jacobien.  $M_k$  est plus facile à calculer et ses propriétés (symétrique et positive) peuvent être choisies convenablement. La relation de Quasi-Newton est donnée par :

$$x_{k+1} - x_k = M_k(f(x_{k+1}) - f(x_k)) \quad (4.2)$$

Les itérations pour ces méthodes deviennent :

$$x_{k+1} = x_k - \alpha_k M_k \cdot f(x_k) \quad (4.3)$$

Avec  $\alpha_k$  un paramètre pour la convergence,  $M_k$  est mise à jour à chaque itération. La mise à jour dépend de la méthode. Les méthodes de Quasi-Newton sont souvent appliquées à la minimisation de fonction  $g(x)$ , exprimée sous la forme  $\nabla g(x)$ . Dans ce cas, c'est la matrice hessienne qui est utilisée.

## 4.2. PREMIÈRE PROPOSITION

BFGS remplace  $\nabla f(x)$  de la fonction à minimiser  $f(x)$  par une matrice hessienne  $H_k$ . La matrice hessienne d'une fonction  $f(x)$  est définie, à condition que toutes les dérivées secondes de  $f(x)$  existent, par  $H(f)$  :

$$H(f) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1^2 \partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_1^2 \partial x_n^2} \\ \frac{\partial^2 f}{\partial x_2^2 \partial x_1^2} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2^2 \partial x_n^2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n^2 \partial x_1^2} & \frac{\partial^2 f}{\partial x_n^2 \partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \quad (4.4)$$

Le hessien est le discriminant de la hessienne.

L'idée de la méthode BFGS est de construire en lieu et place de la matrice hessienne, une approximation de l'inverse de la dérivée seconde de la fonction à minimiser.

*Mise à jour de la méthode BFGS :*

Soit une fonction  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  continue et dérivable,  $x_{k-1}$  et  $x_k$ , tels que  $d_{k-1}^T y_{k-1} > 0$ , avec  $d_{k-1} = x_k - x_{k-1}$  et  $y_{k-1} = \Delta f(x_k) - \Delta f(x_{k-1})$ . Soit une matrice **symétrique définie positive**  $H_{k-1} \in \mathbb{R}^{n \times n}$ .

La mise à jour BFGS est définie par :

$$H_k = H_{k-1} + \frac{y_{k-1} y_{k-1}^T}{y_{k-1}^T d_{k-1}} - \frac{H_{k-1} d_{k-1} d_{k-1}^T H_{k-1}}{d_{k-1}^T H_{k-1} d_{k-1}}$$

Pour minimiser la fonction  $f$  il suffit de résoudre l'équation :

$$H_k d_k = -\Delta f(x) \quad (4.5)$$

L'inverse de  $H_k$  est calculée de la façon suivante :

$$H_k^{-1} = L \cdot H_{k-1}^{-1} \cdot L + \frac{d_{k-1} y_{k-1}^T}{d_{k-1}^T y_{k-1}} \quad (4.6)$$

$$L = \left( I - \frac{d_{k-1} y_{k-1}^T}{d_{k-1}^T y_{k-1}} \right) \quad (4.7)$$

Nous allons maintenant décrire l'algorithme BFGS pour la minimisation d'une fonction donnée, dans notre cas la localisation.

**L'algorithme BFGS appliqué au positionnement** La fonction à minimiser est donnée par :

$$f_i(X) = d_i - \sqrt{(x_i - x)^2 + (y_i - y)^2} \quad (4.8)$$

$d_i$  est la distance entre la position  $(x_i, y_i)$  du nœud de référence  $i$  et la position  $X(x, y)$  de l'utilisateur. Les positions  $(x_i, y_i)$  sont connues pour  $i = 1, \dots, n$ .

La nouvelle position sera estimée en minimisant la fonction par l'Algorithme 2.

## 4.2. PREMIÈRE PROPOSITION

---

Nous allons aussi expérimenter plusieurs schémas d'initialisation de la méthode BFGS afin de comparer, pour un positionnement par méthodes géométriques, si BFGS fait mieux que la combinaison avec une autre méthode. Bien entendu d'autres méthodes Quasi-Newton existent, cependant BFGS est connu pour avoir de meilleures performances en terme de convergence [Dennis et Moré, 1977].

---

**Algorithme 2** : L'Algorithme Broyden-Fletcher-Goldfarb-Shanno pour le positionnement *indoor*

---

- 1: Entrée :
    - La fonction  $f : R^n \rightarrow R^n$ , dérivable ;  $n$  est souvent égal à 2 ou 3.
    - Le gradient de la fonction  $\Delta f : R^n \rightarrow R^n$  ;
    - Une première approximation de la solution  $x_0 \in R^n$  ;
    - Une première approximation de l'inverse du hessien  $H_0^{-1} \in R^{n \times n}$ . par défaut  $H_0^{-1} = I$ .
    - La précision demandée  $\epsilon \in R, \epsilon > 0$ .
  - 2: Sortie : Une approximation de la solution  $x^* \in R$
  - 3: Initialisation de l'algorithme  $x, k \leftarrow x_0, 0$ , Sortir  $\leftarrow$  faux ;
  - 4: **repeat**
  - 5: Calculer  $d_k \leftarrow -H_k^{-1} \Delta f(x_k)$
  - 6: Déterminer  $\alpha_k$  en appliquant la recherche linéaire avec  $\alpha_0 = 1$ .
  - 7:  $x_{k+1} \leftarrow x_k + \alpha_k d_k$
  - 8:  $k \leftarrow k + 1$
  - 9: Mettre à jour  $H_k^{-1}$  :
$$H_k^{-1} = \bar{L} \cdot H_{k-1}^{-1} \cdot \bar{L} + \frac{\bar{d}_{k-1} y_{k-1}^T}{\bar{d}_{k-1}^T y_{k-1}}$$
Avec  $\bar{L} = \left( I - \frac{\bar{d}_{k-1} y_{k-1}^T}{\bar{d}_{k-1}^T y_{k-1}} \right)$ ,  $\bar{d}_{k-1} = \alpha_{k-1} d_{k-1} = x_k - x_{k-1}$  et  $y_{k-1} = \Delta f(x_k) - \Delta f(x_{k-1})$
  - 10: **if**  $\|\Delta f(x_k)\| \leq \epsilon$  **then**
  - 11:      $x^* \leftarrow x_k$
  - 12:     Sortir  $\leftarrow$  vrai
  - 13: **end if**
  - 14: **until** Sortir  $\leftarrow$  vrai
- 

### 4.2.2 Le Réseau de Capteurs

#### 4.2.2.1 Description

Le réseau est composé de 11 capteurs Waspote [Libelium, 2012] avec des modules Zigbee appelés XBEE ZB [Digi Inc, 2013] opérant dans la bande de fréquence 2.4 GHz. XBEE est le module contenant le *transceiver* radio ainsi que l'antenne. Il est fabriqué par la société Digi et se décline en plusieurs versions. XBEE ZB implémente le protocole Zigbee. Et c'est cette dernière que nous avons utilisée. La création d'un réseau Zigbee passe en grande partie par la configuration des modules XBEE afin de leur assigner les rôles dans le réseau.

## 4.2. PREMIÈRE PROPOSITION

---

Model	Protocol	Frequency	txPower	Sensitivity	Range *
XBee-802.15.4	802.15.4	2.4GHz	1mW	-92dB	500m
XBee-802.15.4-Pro	802.15.4	2.4GHz	100mW	-100dBm	7000m
XBee-ZB	ZigBee-Pro	2.4GHz	2mW	-96dBm	500m
XBee-ZB-Pro	ZigBee-Pro	2.4GHz	50mW	-102dBm	7000m
XBee-868	RF	868MHz	315mW	-112dBm	12km
XBee-900	RF	900MHz	50mW	-100dBm	10km
XBee-XSC	RF	900MHz	100mW	-106dBm	12km

\* Line of sight and 5dBi dipole antenna



Figure 3: XBee

FIGURE 4.3 – Les valeurs de l’antenne XBEE.



FIGURE 4.4 – Coordinateur Waspote avec son module XBEE.

Nous avons configuré les modules afin d’avoir 1 coordinateur et 10 routeurs. Ces capteurs ont aussi des modules GSM/GPRS et Bluetooth ainsi que des capteurs de température, de luminosité, d’humidité et un accéléromètre.

La topologie actuelle du réseau Zigbee est une topologie étoile centrée sur le coordinateur.

### 4.2.2.2 Le coordinateur

Le coordinateur est responsable de la création et de la configuration du réseau (ce qui inclut scanner les canaux disponibles) ; de donner accès aux routeurs cherchant à se rattacher à un réseau et d’envoyer les distances obtenues à partir des routeurs au Superviseur et au Smartphone.

La Figure 4.4 donne une illustration d’un noeud Waspote servant de coordinateur et de pont entre le PC du superviseur et le réseau.

Lorsque le coordinateur reçoit une trame d’un noeud, il détermine la cellule correspondante et remplit la ligne correspondante à la cellule avec la distance et l’identifiant du noeud tant que les distances reçues ne sont pas au nombre de trois. Le coordinateur peut aussi remplacer dans le tableau *Data*, une entrée correspondant à une cellule par la distance la plus courte. Une fois le tableau rempli, le coordinateur l’envoie au superviseur et au smartphone. Le fonctionnement du coordinateur est résumé par l’Algorithme 3.

---

**Algorithme 3** : Fonctionnement du Coordinateur

---

- 1: Initialisation : Création du réseau Zigbee
    - Choix du PANID
    - Choix du Canal
    - Choix de la clé AES si besoin
  - 2: Démarrer le *transceiver* et la liaison USB
  - 3: **repeat**
  - 4: Écouter les transmissions Zigbee
  - 5: Pour Chaque routeur  $r_i$
  - 6: Pour chaque trame reçu :
    - Déterminer l'ID de la cellule et l'identifiant
    - Remplir la cellule  $i$  du tableau correspondant au routeur  $r_i$  si elle est vide ou si elle est mieux que la précédente. « Mieux » signifie plus courte distance.
  - 7: Retourner le tableau des données
  - 8: **until** Critère d'arrêt
- 



FIGURE 4.5 – Routeur Wasmote avec un transceiver Bluetooth.

#### 4.2.2.3 Les routeurs

Les routeurs déterminent les informations concernant le smartphone, lorsqu'il est dans un rayon atteignable en utilisant leur module GSM/GPRS.

La Figure 4.5 donne une illustration d'un noeud Wasmote servant de routeur.

Après avoir déterminé l'identifiant de cellule du smartphone ; les distances sont estimées par TOA et RSSI en utilisant les commandes fournies par l'API des capteurs Wasmote. Après cela, le routeur envoie une trame contenant  $CellID_i, id_i, d_i$  au coordinateur. Comme illustré par la figure 4.1.

Les questions que ce choix implique sont : est-ce que le système fonctionnera pour chaque opérateur simplement indépendamment de la carte SIM utilisée ? Avons-nous besoin d'enregistrer les smartphones ? Si oui par quel identifiant ? Un numéro de téléphone suffit-il ?

**Méthodes d'estimation de la distance** Nous rappelons ici que les méthodes géométriques sont basées sur la distance et que comme cela a été montré dans le chapitre 2, l'estimation de la distance souffre des perturbations dans la zone de déploiement du réseau. L'architecture que nous avons proposée est donc à évaluer en prenant en compte les difficultés liées à ces perturbations. Pour l'évaluation, nous avons proposé :

**Algorithme 4** : Fonctionnement d'un routeur

- 
- 1: Initialisation : Se connecter au réseau Zigbee
    - Chercher un coordinateur
    - Récupérer les informations
  - 2: Démarrer les *transceiver*, Zigbee et GSM dans notre cas
  - 3: **repeat**
  - 4: Scanner les appareils GSM disponibles
  - 5: Déterminer le smartphone
  - 6: Estimer les distances par TOA et RSSI.
  - 7: Calculer une moyenne si besoin.
  - 8: Envoyer les trames au Coordinateur ;
  - 9: **until** Critère d'arrêt
- 

- une simple table de correspondance distance/rssi ;
- génération de données aléatoires ;
- calibration d'un modèle de propagation *indoor*.

Une méthode simple de RSSI, est la suivante, à partir des mesures RSSI reçues pour plusieurs points, nous allons associer une distance à chaque ensemble de mesure comme indiqué par le Table 4.1. Un ensemble de mesures peut tourner autour d'une valeur moyenne ou d'un mode déterminé dans les mesures. Chaque table est à stocker sur le nœud mobile qui fera l'association en fonction des distances mesurées.

RSS (dBm)	Distance (m)		
	$n_1$	$n_2$	$n_3$
-54	1	2	3
-53	1.4	3.4	5
-60	2.8	1	5

TABLE 4.1 – Table d'association simple de RSSI et de Distance pour un point de référence.

La seconde approche consiste à générer des données aléatoires suivant plusieurs scénarios. Nous avons déjà décrit BonnMotion [BonnMotion Team, 2013] qui permet la génération de données pour la simulation. Les scénarios comprendront aussi bien les chambres, l'empiétement des chambres que la génération de données pour un chemin suivi par l'utilisateur.

La troisième approche consiste à calibrer un modèle de propagation pour l'environnement et déterminer la distance à partir de la puissance ou du temps de vol.

Pour la relation RSSI-Distance c'est le modèle suivant que nous allons explorer :

$$P_r = P_0 - 10 \times \eta \log\left(\frac{d}{d_0}\right) + X_\sigma \quad (4.9)$$

Avec  $P_r$  le niveau de la puissance reçue,  $d_0$  est une distance arbitraire généralement choisie comme 1 m,  $\eta$  est le facteur de la propagation et  $X_\sigma$  est la variable aléatoire prenant en compte les effets de masque.

Pour un nœud Wasmote, le RSSI d'un paquet reçu peut se déterminer par simple appel de fonction. Il nous faut déterminer quel est l'impact de la sensibilité du récepteur sur la valeur reçue.

### 4.2.3 Le Smartphone

Le smartphone utilisé dans ce prototype est un Nexus One sous Android. Il constitue le terminal mobile qui va afficher la position à l'utilisateur et aussi d'autres informations concernant le milieu environnant comme les salles de cours aux alentours, la bibliothèque, etc.

Les algorithmes de positionnement sont les mêmes que ceux décrits pour l'application Superviseur. Notons que les capteurs embarqués dans le smartphone tels l'accéléromètre, la photodiode, la caméra et le microphone peuvent être exploités pour enrichir la position.

Une fois la position déterminée, nous allons simuler plusieurs scénarios de déplacement à l'aide de BonnMotion (marche lente, évacuation, etc.) nécessitant la recherche de chemins. Celle-ci peut être faite de plusieurs manières. Nous avons utilisé l'algorithme de Dijkstra.

Un point crucial est la communication entre le smartphone et le réseau de capteurs. Ceci n'est envisageable, au regard de la technologie des capteurs Wasmote à notre disposition au moment du test, qu'à travers le gateway Wasmote. Cependant, celui-ci ne communique qu'avec du Zigbee. De ce fait, nous avons envisagé un échange de données par liaison WiFi ou Bluetooth. Ce qui pour un smartphone induit un coût en énergie.

Dans la section suivante nous allons présenter la mise en œuvre de ce prototype.

### 4.2.4 Expérimentations

Les expériences ont été envisagées de deux façons, une première est le test des méthodes avec des données artificielles générées par des gaussiennes. L'objectif est de pouvoir modéliser et simuler la propagation des ondes d'un réseau Zigbee. Nous allons présenter des modèles générales. Une autre partie décrira les tentatives de mises en œuvre pratique du système, des problèmes que nous avons rencontrés.

La figure 4.6 montre comment nous allons procéder :

- une simple méthode suffira-t-elle et dans quelles circonstances ?
- dans quels cas faut-il utiliser une méthode de raffinement de la position et comment l'initialiser ?

Dans la suite nous allons détailler ces points.

#### 4.2.4.1 Description des données

Les données sont générées par des lois gaussiennes, avec comme hypothèse que les paramètres des lois reflètent les conditions de propagation. Nous allons ajouter un bruit blanc gaussien au distance. Soit un point sur le plan de l'environnement de positionnement fourni par une image.

Le cas le plus simple est de considérer trois nœuds de référence dont les puissances sont reçues dans chaque chambre. Ces trois nœuds de référence forment un repère. Un autre cas,

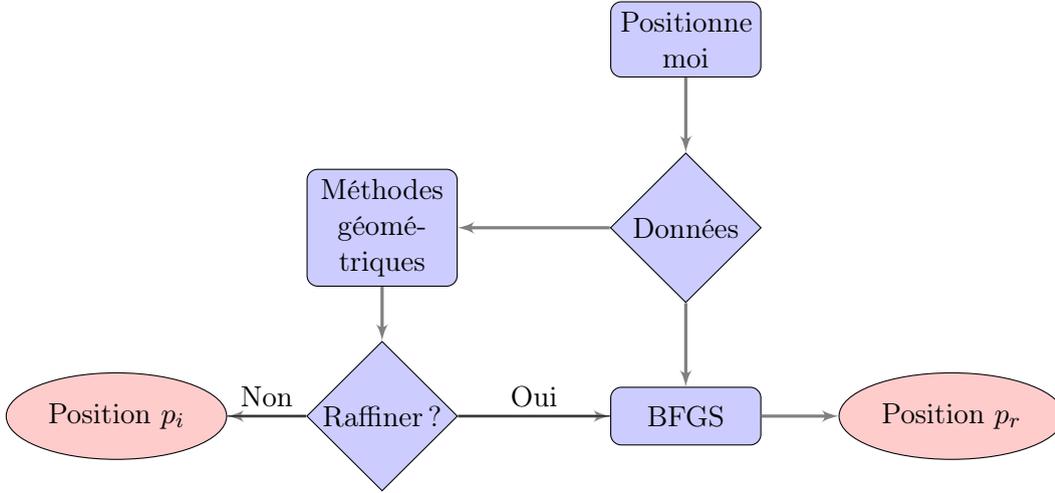


FIGURE 4.6 – Illustration du positionnement par notre prototype. À l’initialisation des expérimentations nous pouvons d’abord déterminer une position par une des méthodes géométriques et vérifier la précision obtenue par cette méthode. Nous pouvons aussi décider d’améliorer cette précision en la raffinant par la méthode BFGS. Une autre possibilité est de directement déterminer une position par la méthodes BFGS en étudiant les conditions d’initialisation et de convergence.

plus compliqué, est d’utiliser un *overlapping* entre les chambres. Ce qui veut dire introduire une incertitude pour certains points quant à leur appartenance à telle ou telle partie de la chambre ou à telle chambre.

Les données utilisées dans ces expérimentations, sont générées pour le temps d’arrivée et la puissance du signal en considérant deux cas de figure : LOS et NLOS.

### Modèle de données RSSI

**Le cas LOS** L’équation 4.9 décrit la relation qui lie puissance reçue au niveau d’un récepteur, la distance parcourue et les effets de masques.  $X_i$  suit une loi *log*-normale de moyenne nulle et d’écart-type  $\sigma_{X_i}$ . La loi suivie par la puissance  $P_i$  est donnée par :

$$P_i \sim \log N\left(P_0 - 10\eta \log\left(\frac{d_i}{d_0}\right), \sigma_{X_i}^2\right) \quad (4.10)$$

Afin de pouvoir générer des données avec ce modèle, il faudra déterminer les valeurs de  $P_0$ ,  $d_0$ ,  $\eta$  et  $\sigma_{X_i}$ . Nous pouvons, alors, générer aléatoirement des puissances, en connaissant les positions des deux nœuds.

Ce modèle peut aussi nous servir à générer des bases de données pour une simulation des méthodes de *fingerprinting*. Il suffit de considérer un ensemble de lois que nous venons de décrire. Par exemple :

- à la phase de calibration nous générons une base de données qui contient  $m$  vecteurs :

$$\mathbf{rssi} = [P_1, P_2, \dots, P_m] \quad (4.11)$$

## 4.2. PREMIÈRE PROPOSITION

---

Avec  $P_i$  donnée par l'équation 4.10 et  $m$  est le nombre de points de références. Notons, comme nous l'avons déjà dit que d'autres lois peuvent être utilisées, comme une loi de Weibull. Il peut aussi être intéressant d'étudier le cas de vecteurs composés de plusieurs types de loi.

- la phase de positionnement se basera sur un autre vecteur généré par le même modèle et à une position quelconque. Le modèle basé sur la puissance donne la puissance qu'un récepteur est sensée recevoir à une position donnée au regard des conditions de propagation. Pour plus de réalisme, on peut s'intéresser au récepteur et prendre en compte la perte de puissance liée aux circuits du récepteur. Cependant nous n'avons pas étudié dans quel mesure cela peut influencer le modèle.

L'enjeu sera de savoir si la méthode peut déterminer la position et avec quelle erreur.

Une autre approche consiste à déterminer un estimateur de la distance émetteur-récepteur à partir de la puissance, considérée, comme étant connue, par exemple, une valeur déterminée après une campagne de mesure ou la puissance effective (voir Chapitre 1). L'équation 4.12 nous donne cet estimateur.

$$\tilde{d}_i = d_0 10^{\frac{P_i - P_0}{10\eta}} = d_i \exp\left(\frac{X_i \ln 10}{10\eta}\right) \quad (4.12)$$

$\tilde{d}_i$  suit une loi normale  $N(\mu, \sigma^2)$  avec :

$$\mu = d_i \exp\left(\frac{1}{2} \left(\frac{\sigma_{x_i} \ln 10}{10\eta}\right)^2\right) \quad (4.13)$$

$$\sigma^2 = d_i^2 \exp\left(\frac{\sigma_{x_i} \ln 10}{10\eta}\right)^2 \left(\exp\left(\frac{\sigma_{x_i} \ln 10}{10\eta}\right)^2 - 1\right) \quad (4.14)$$

À partir de l'équation 4.13, nous pouvons remarquer que cet estimateur est biaisé puisque  $\mu \neq d_i$ . Pour enlever le biais, il suffit de diviser  $\tilde{d}_i$  par  $\exp\left(\frac{1}{2} \left(\frac{\sigma_{x_i} \ln 10}{10\eta}\right)^2\right)$ . L'estimateur sans biais  $\bar{d}_i \sim N(\mu, \sigma^2)$  avec :

$$\bar{d}_i = \tilde{d}_i \exp\left(-\frac{1}{2} \left(\frac{\sigma_{x_i} \ln 10}{10\eta}\right)^2\right) = d_i \exp\left(\frac{X_i \ln 10}{10\eta}\right) \exp\left(-\frac{1}{2} \left(\frac{\sigma_{x_i} \ln 10}{10\eta}\right)^2\right) \quad (4.15)$$

$$\mu = d_i \quad (4.16)$$

$$\sigma = d_i^2 \left(\exp\left(\frac{\sigma_{x_i} \ln 10}{10\eta}\right)^2 - 1\right) \quad (4.17)$$

Une fois la distance déterminée, nous pouvons appliquer une des méthodes géométriques pour estimer la position.

### Remarques sur le modèle

1. L'erreur du second, cas,  $e_i = \bar{d}_i - d_i$  est alors une variable *log*-normale de moyenne nulle et d'écart-type  $\sigma_i$  ;
2. Le CRLB sur  $P_i$  est donnée par :

$$\sigma_i \geq \frac{\ln 10 \sigma_{x_i} d_i}{10\eta} \quad (4.18)$$

Nous pouvons alors constater que la précision en utilisant ce modèle dépend du facteur de propagation  $\eta$ , de la distance  $d_i$  et aussi de l'écart-type de  $\sigma_{x_i}$  donc des effets de masques. On voit bien que :

- en augmentant  $\eta$ , l'erreur diminue. Mais qu'elle sera l'effet de cette augmentation sur la propagation ? Nous savons (Voir Chapitre 1) que des valeurs élevées en milieu intérieur symbolise une guide de l'onde.
- le fait de diminuer la distance  $d_i$ , à toutes valeurs égales, ferait baisser l'erreur. Cela signifie qu'il faudrait augmenter le nombre de points de référence. Cependant avons nous suffisamment de points de références pour ce faire ? Quel en sera les coûts de déploiement et de maintenance ?
- minimiser  $\sigma_{x_i}$ . Le contrôle des effets de masques sera très difficiles à réaliser. Nous savons aussi que des valeurs élevé auront l'effet de baisser la puissance du signal reçue et du coup d'affaiblir la qualité de l'estimateur de la distance.

**Le cas NLOS** peut être considérée de plusieurs manières. D'abord en supposant que le modèle, donnée précédemment, prend aussi en compte les effets du multitrajets. On peut aussi utiliser un modèle comme Rice ou Nagakami (Voir Chapitre 1).

### Modèle de données TOA

**Le cas LOS** est considéré lorsque le signal se propage en ligne directe. Dans ce cas, il ne subit que les effets du bruit blanc gaussien et les effets du multitrajet.

Soient  $t_i$  le temps d'émission et  $t_j$  le temps de réception, la distance parcourue est estimée par :

$$\tilde{d}_i = c(t_j - t_i) = d_i + e_i \quad (4.19)$$

Avec  $d_i$  la vraie valeur de la distance et  $e_i$  l'erreur suivant une loi  $N(0, \sigma_i^2)$ . Ainsi la distance :

$$\tilde{d}_i \sim N(d_i, \sigma_i^2) \quad (4.20)$$

En considérant le canal comme gaussien (c'est-à-dire qu'il subit un bruit blanc gaussien), la borne inférieure de Cramer et Rao sur la variance  $\sigma_i$  est donnée par :

$$\sigma_i \geq \frac{c}{2\sqrt{2\pi}\gamma\beta} \quad (4.21)$$

Avec  $\gamma$  le SNR et  $\beta$  la bande passante effective.

#### Remarques sur le modèle

1. La variance est inversement proportionnelle à la bande passante du signal. Ce qui signifie qu'une large bande permet d'atteindre des faibles valeurs du CRLB. C'est la raison pour laquelle, l'*Ultra Wide Band* [Gezici *et al.*, 2005] atteint une précision centimétrique. Le même niveau de précision a été rapporté pour des ondes radios et acoustique [Yan, 2010].
2. Les paramètres de la loi ( $\mu$  et  $\sigma$ ) sont constants pour tous les chemins LOS. Cependant pour certains auteurs, ces paramètres sont fonctions de la vraie distance et de la bande passante. Ceci s'explique par [Yan, 2010] :

- le fait que la résolution d’une technique de mesure TOA dépend de la bande passante ;
- la probabilité de séparation spatiale entre le chemin direct et le multitrajet à la réception a tendance à décroître lorsque la vraie distance entre l’émetteur et le récepteur augmente.

Ainsi, le modèle donnée dans 4.20 est raffinée par :

$$\tilde{d} \sim N(\mu(\beta, d_i), \sigma_i^2(\beta, d_i)) \quad (4.22)$$

Où  $\beta$  est en GHz .  $\mu(\beta, d_i), \sigma_i^2(\beta, d_i)$  peuvent s’exprimer de plusieurs manières. La détermination de ces fonctions dépend de l’environnement, de la norme de transmission et de la technologies des nœuds utilisés dans la campagne de mesure.

**Le cas NLOS** considère que le signal ne se propage pas en ligne directe. Dans ce cas nous avons :

$$\tilde{d}_i = d_i + \xi_i + e_i \quad (4.23)$$

Où  $d_i$  est la vraie distance séparant l’émetteur du récepteur,  $\xi_i$  est une variable aléatoire qui suit une loi de probabilité de moyenne non nulle et représente le biais introduit par les chemins NLOS, et  $e_i \sim N(0, \sigma^2)$  est comme dans le cas précédent.

On distingue deux cas de figure :

1.  $\xi_i$  est traité comme une variable déterministe inconnue dans le système d’équations. Il est déterminé par la méthode de positionnement et il faut vérifier que le système d’équations est solvable (*i.e* qu’on a le même nombre d’équations que de variables). Ainsi le modèle devient :

$$\tilde{d} \sim N(d_i + \xi_i, \sigma_i^2) \quad (4.24)$$

2.  $\xi_i$  est traité comme une variable aléatoire suivant une loi de probabilité. La loi exponentielle est la plus utilisée dans ce cas [Yan, 2010]. Il faut noter qu’une calibration est à faire comme dans le cas précédent.

Nous venons de présenter des modèles pour la génération de données pour des mesures TOA et RSSI. Cependant, il est possible d’établir des modèles pour les autres techniques de mesure.

### 4.2.4.2 Description de l’environnement de positionnement

L’environnement de l’expérimentation est donné par le plan illustré par la Figure 4.7. Les raisons de ce choix sont :

1. tester un repère globale ;
2. tester un positionnement pour chaque salle ;
3. déterminer une façon de gérer les positions entre deux salles ;
4. déterminer les positions dans les couloirs ;
5. générer des scénarios d’évacuation (voir BonnMotion) ;
6. générer des données pour un *fingerprinting*.

Nous allons décrire ces cas.

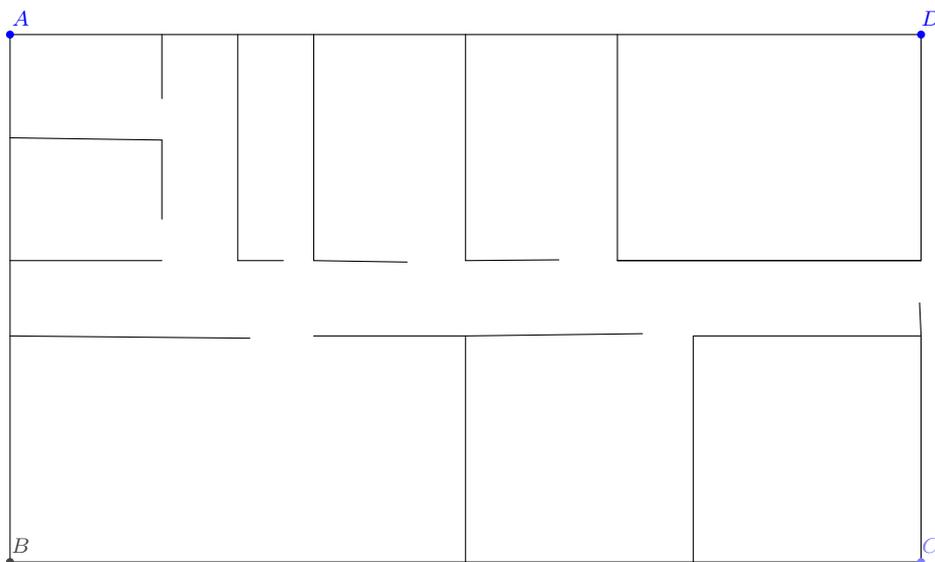


FIGURE 4.7 – Exemple de plans pouvant être utilisés pour tester notre prototype. Nous donnons ce plan sans les dimensions par souci de simplicité.

**Cas 1** Le premier cas consiste à prendre le rectangle ABCD et à définir un repère avec les points A, B, C. Au niveau de ces points, nous considérons que des nœuds de référence sont déployés avec pour hypothèse que les portées de ces nœuds couvrent toute la surface du rectangle ABCD. En d'autres termes, en tous points de ce rectangle, nous recevons un signal de chaque nœud de référence (Hypothèse valable en 3D). Le choix de la dimension (2D ou 3D) pour les positions relève plus des besoins de l'application. Un positionnement en 2D est une projection de la position dans un des plans.

Avant de continuer, nous allons soulever deux problèmes :

1. quel est le meilleur déploiement des nœuds de références ? Pour répondre à cette question, nous allons analyser plusieurs schémas de déploiement. Cependant, dans le cadre de cette thèse nous n'avons pas approfondi les diagrammes de propagation des antennes au point de déterminer quel déploiement offrira le plus de couverture dans notre environnement. Cependant nous orientons le lecteur vers la thèse de Priyantha [Priyantha, 2005] qui a étudié cette question pour une onde acoustique.
2. comment et où l'utilisateur devra-t-il porter son nœud ? Et quelle granularité choisit-on ? Ce choix est lié à la précision, par exemple, si nous choisissons une précision de niveau pièce, l'utilisateur peut simplement être positionné indifféremment sur dans la salle considéré comme son bureau. Si nous choisissons un niveau sous-pièce, pourrions-nous déterminer s'il est sur son bureau sachant que son smartphone (cet appareil étant considéré comme le nœud à positionner) est dans sa poche ou simplement posé sur le bureau ? Et comment faire pour savoir si l'utilisateur porte son smartphone ? Remarquons que ce problème est lié à la sémantique que l'on veut donner à une position (Voir Chapitre 2).

Considérons la Figure 4.8. Cette figure avec les dimensions d'un bâtiment de 45 mètre

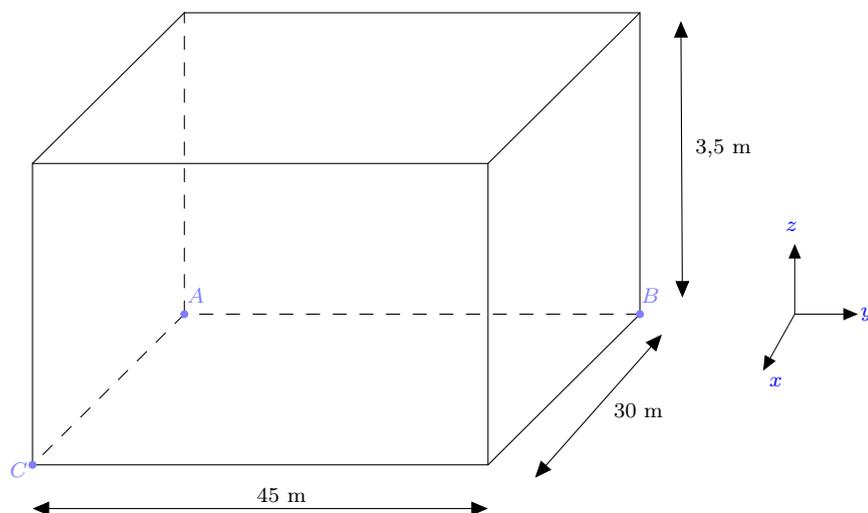


FIGURE 4.8 – Illustration du plan avec les dimensions.  $A(0,0,0)$ ,  $B(0,45,0)$ ,  $C(30,0,0)$  sont des point de référence définissant le repère. Cependant nous aurions pu choisir un déploiement différent.

de longueur, 30 mètre de largeur et 3,5 mètre de hauteur va nous servir à illustrer les divers cas. Le choix des point de référence, est trivial, dans cet exemple. Mais il sera possible de choisir d'autre coordonnées. Le premier problème que nous avons soulevé, concernant les points de référence se pose à ce niveau. En réponse, les coordonnées choisies devront prendre en compte les besoins de l'application en terme de précision et de ce fait doivent être les positions qui dans le bâtiment offre cette couverture.

Le but de ce cas de figure est de choisir des points appartenant à différentes salles du bâtiment et de faire déterminer par notre prototype les positions dans le repère globale. Les données seront générées de la façon suivante. Soit la position  $p$  de coordonnées  $(x, y, z)$ . Nous allons définir une variable aléatoire  $\tilde{d}$  suivant une loi normale  $N(\mu, \sigma)$  dont les paramètres sont définis de la façon suivante :  $\mu = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2}$ , avec  $(x_i, y_i, z_i)$  les coordonnées du point de référence  $i$  ; et  $\sigma$  sera choisi entre 0 et 1.

**Cas 2,3,4** Il s'agit, dans le second cas, de définir un repère pour chaque salle avec un schéma de déploiement qui différent. Une fois que le déploiement est effectué, la génération des données est la même que celle décrite précédemment. Dans ce second cas, notre objectif est d'étudier les erreurs de positionnement pour une même position en fonction de plusieurs déploiement de nœuds dans les deux types de repères : globale et locale.

Le troisième cas se pose la question de comment déterminer les positions situées à côtés des frontières entre les salles.

Le quatrième cas détermine la position dans les couloirs. Nous pouvons considérer un couloir comme une salle en la délimitant par des points de références. Même si sur le plan théorique cela est faisable, dans la réalité cette solution est difficilement applicable du fait notamment de la propagation. Nous pouvons aussi découper le couloir en petites zones dont chacune ayant ses points de références à la condition que nous en disposons suffisamment.

## 4.2. PREMIÈRE PROPOSITION

---

Dans le cas d'un positionnement globale, il est assez facile de déterminer la position dans une zone donnée du bâtiment. Pour cela le plan est délimité en zone dont les coordonnées serviront à déterminer le lieu en question. Cependant ce découpage se fait au niveau de l'application.

**Le cas 5** Il s'agit ici, de tester des algorithmes de détermination du plus court chemin. Pour cela, pour un scénario donné (évacuation d'urgence, visite guidée, etc), de générer le chemin pas par pas. Le pas sera défini par le scénario par exemple, on peut considéré que chaque point du chemin est séparé par une distance de 0,5 mètre et que la personne se déplace en ligne droite dans chaque espace. Nous renvoyons le lecteur à la documentation de BonnMotion [Aschenbruck *et al.*, 2010].

**Le cas 6** Ici, nous voudrions générer des données pour les algorithmes classification ou de *fingerprinting*. Chaque salle sera considérée comme une classe aussi bien dans un repère globale que locale. Nous allons choisir un certain nombre de positions pour générer le *fingerprint*. C'est le modèle donné par l'équation 4.12 que nous allons utiliser. Nous allons modéliser une salle par :

- son ensemble de points de référence et pour chacun de ses nœuds la puissance d'émission ;
- le ou les facteurs de propagation. Le cas simple est de considérer la même valeur pour toute la salle

Par exemple, en considérant un réseau Zigbee constitué de nœuds émettant dans la bande de fréquence 2.4 GHz avec une puissance d'émission de 1mW dans un rayon de 1 m. Pour convertir cette puissance en dBm nous utilisons la formule :

$$P(dBm) = 10 \log(P/1mW) \quad (4.25)$$

La puissance est donc de 0dBm.

- En, nous référant à la section 1.6.6.1 du Chapitre 1 nous avons les valeurs suivant pour :
- $\eta = 7$ , valeur qui correspond à une salle ;
  - $\sigma = 1.6$  qui est la valeur minimale pour une propagation en milieu interne en ligne directe.

Avec ces valeurs, nous pouvons définir une variable aléatoire suivant une loi normale

$$N(P_0 - 10\eta \log(d/d_0), \sigma^2)$$

qui avec les valeurs choisies devient :

$$N(-70 \log(\|x - x_i\|), 2, 56) \quad (4.26)$$

Où  $\|x - x_i\|$  est la distance euclidienne entre la position connue du nœud et la position du point de référence. La distance euclidienne nous donne la vraie valeur de la distance entre les deux nœuds. Comme nous l'avons vu à plusieurs reprise cette distance n'est pas connue pour une application réelle et les modèles que nous avons décrits permettent de trouver un estimateur. Dans tous les cas de figure que nous avons abordés pour les modèles

### 4.3. SECONDE PROPOSITION

---

de données, les modèles doivent prendre en compte les facteurs rendant cette vraie valeur de la distance indéterminable et y permettre de trouver un estimateur assez proche.

Ainsi, si nous avons 3 points de référence dans la salle, nous pouvons générer une base de donnée pour un nombre de point choisis par :

$$N(-70 \log(\|x - x_i\|), 2, 56), i = 1..3 \quad (4.27)$$

Cependant pour plus de réalisme, on peut intéresser à des valeurs différentes pour les paramètres de la loi.

#### 4.2.4.3 Fonction d'évaluation ou mesure de précision

la fonction d'évaluation est l'erreur moyenne entre la position connue et la position trouvée par le système. Nous allons aussi nous servir de l'écart-type.

$$e_i = \tilde{d}_i - d_i \quad (4.28)$$

## 4.3 Seconde proposition

Nous allons nous servir de tous types d'appareils capable d'émettre et de recevoir pour localiser et guider la personne handicapée dans un bâtiment. Ainsi nous étendons la définition d'un nœud à :

- un nœud d'un réseau de capteurs ;
- smartphone, PDA, PC, spot WIFI, etc ;
- le matériel électroménager : TV, Réfrigérateur, Micro-onde, etc. ;
- puces RFID, Camera, etc ;
- fauteuils roulants intelligents.

Certains de ces nœuds sont mobiles, portés par la personne handicapée alors que d'autres sont fixes et servent de nœud de référence. Le nœud mobile peut aussi servir de point de référence une fois sa position déterminée. Nos objectifs sont de proposer des solutions pour :

- Les personnes handicapées en prenant en compte leurs défis quotidiens et le fait que l'environnement doit être tolérant aux pannes et accessible ;
- la localisation des nœuds fixes et mobiles ;
- Incorporer une intelligence dans l'environnement en utilisant les méta-heuristiques dites Intelligence Collective.

L'architecture du système reprend et approfondie celle du prototype précédent. Elle se subdivise en trois parties importantes : la couche réseau, le middleware et le système de localisation.

#### 4.3.1 La Couche Réseau

De nos jours, plusieurs réseaux existent dans un bâtiment. Cette hétérogénéité peut être exploitée afin de fournir un réseau plus dense. La question sera d'étudier comment cette forte densité contribue à augmenter la précision.

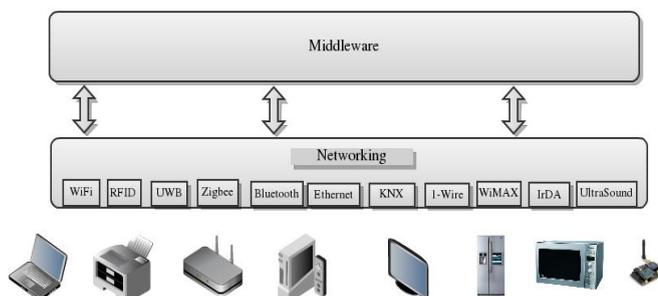


FIGURE 4.9 – La couche réseau

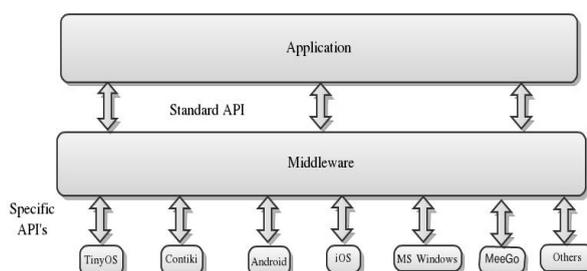


FIGURE 4.10 – Le middleware

Nous considérons les réseaux filaires ou non pour réduire les coûts, faciliter le déploiement et garantir la robustesse. Avant de déployer notre système, une étude de l'environnement sera menée pour prendre en compte les réseaux déjà existants, le plan du bâtiment et les besoins spécifiques à traiter. Ainsi, en considérant le positionnement de certains nœuds, nous allons déterminer un déploiement de nœuds supplémentaires et/ou le redéploiement de certains nœuds. Par exemple changer l'emplacement de la télévision afin de mieux localiser la personne dans le salon. Notre objectif a été d'étudier le phénomène de la propagation dans le département Polytech'Tours par simulation. Pour cela, une proposition est de découper la zone de déploiement en petit celle hexagonale et d'appliquer la méthode des éléments finis pour la simulation.

La Figure 4.9 illustre la couche réseau.

La couche réseau doit aussi gérer les interférences, la sécurité des transmissions (un-saut et multi-sauts), la sauvegarde de l'énergie et le contrôle de la topologie.

### 4.3.2 Le Middleware

Le middleware permettra l'indépendance envers une technologie donnée. A terme, il permet à plusieurs matériels différents avec des systèmes d'exploitation différents de communiquer de façon transparente. La Figure 4.10 montre le rôle charnière du middleware à savoir : i) procurer de la transparence en termes de distributivité et d'hétérogénéité des ressources ; ii) fournir une API standard pour les applications et les développeurs ;

Le middleware devrait être léger pour pouvoir être facilement déployé sur divers nœuds. Notre intention a été d'intégrer dans le middleware les méthodes de positionnement que nous avons abordées dans le Chapitre 2.

### 4.3.3 Localisation

Ce sous-système fournit à l'utilisateur les services de localisation. La première fois qu'un utilisateur entre dans la zone de déploiement, le type et le système d'exploitation de son appareil seront déterminés. Il lui sera proposé de se connecter à un serveur, de télécharger et installer l'application sur son nœud. Ensuite, il choisira la politique de localisation.

Une politique de localisation est l'ensemble des mesures mises en place pour assurer le service de localisation en préservant la vie privée de l'utilisateur. Nous avons choisi de fournir deux types de services :

- la localisation passive : la détermination de la position est faite par le nœud au niveau local sans qu'aucune information ne soit envoyée au niveau central. Toutes fois avec l'accord de l'utilisateur certaines informations peuvent être transmises, par exemple sa position courante.
- la localisation active : c'est le système qui calcule et enregistre les déplacements de l'utilisateur dans une base de données. Ce cas est mis en place pour des raisons de sécurité. L'utilisateur sera traqué lorsqu'il entre dans certaines zones réservées et informé de cela.

Ainsi, après détermination de la position, le système fournira à l'utilisateur une assistance à la mobilité.

L'assistance à la mobilité est le processus qui détermine un chemin à partir de la position courante du nœud vers une autre position en prenant en compte les contraintes et en évitant les obstacles. Nous avons considéré un graphe  $G(V, E)$  où  $V$  est l'ensemble des positions et  $E$  les relations entre ces positions (distance, énergie, etc.). Ainsi, la détermination d'un chemin peut être faite : i) en calculant le plus court chemin par l'algorithme de Dijkstra, etc ; ii) ou dans un cas multi-objectifs, un chemin minimisant ou maximisant un ensemble de critères sera déterminé. Des techniques de robotique mobile seront aussi utilisées pour éviter les obstacles qui peuvent survenir de façon aléatoire puisque l'environnement est dynamique.

Dans cette partie, nous comptons modéliser le comportement de l'entité en utilisant par exemple un filtre bayésien.

Ainsi, nous avons défini un certain nombre de profils d'utilisateurs concernant l'assistance à la mobilité :

- Un utilisateur avec les trois handicaps (moteur, visuel et auditif) nécessite de ce fait la présence d'un opérateur humain.
- Un utilisateur ayant un handicap moteur et visuel sera doté d'un fauteuil roulant intelligent équipé par exemple d'un terminal braille et des écouteurs pour transmettre et recevoir les instructions. Cet utilisateur doit être suivi avec une grande précision.
- Un utilisateur ayant un handicap moteur et auditif sera équipé, par exemple, par un fauteuil roulant intelligent muni d'un écran tactile pour les instructions.

- Un utilisateur ayant un seul des handicaps, peut soit utiliser un fauteuil roulant intelligent ou son appareil, par exemple un smartphone.

## 4.4 Conclusion

Ce chapitre a présenté nos travaux sur le positionnement à l'intérieur des bâtiments.

Dans un premier, la question importante ont été d'expérimenter rapidement les méthodes de positionnement. Pour ce faire nous avons proposé une nouvelle méthode de positionnement basée sur la combinaison du Zigbee et du GSM. Nous avons aussi voulu expérimenter plusieurs schéma d'initialisation pour les méthodes dites de raffinement de la position. A savoir si la phase de raffinement améliore ou pas la position fournie par le système. La méthode BFGS a été testée en combinaison d'autres méthodes géométriques.

Dans un second temps, nous avons proposé une autre solution à base de middleware. Cette approche a pour avantages de répondre aux défis posés par la diversité des technologies de positionnement. Et aussi d'un point de vue applicatif, elle offre aux développeurs une API des algorithmes de positionnement.

Enfin nous avons modélisé un environnement *indoor* comme un ensemble de cellules. Dans ces cellules, nous avons étudié les phénomènes de propagation des ondes électromagnétiques et de simuler le positionnement.

Il ressort que les propositions que nous avons effectués sont innovantes et prometteuses. Elles ont été l'objet d'une publication scientifique [Amadou Kountché *et al.*, 2011]. Nous avons aussi effectué une troisième proposition qui visait à utiliser la simulation par éléments finis. Pour cela nous avons proposer de modéliser l'environnement *indoor* en petites cellules, comme dans un réseaux cellulaire, et pour chaque cellules de considérer les matériaux et les phénomènes de propagation.

Nous allons maintenant changer de registre en passant à la partie consacrée à la classification automatique de données.

## Troisième partie

# Étude et Application de la classification par algorithmes de fourmis



## Chapitre 5

# La classification automatique de données

*Mme Laborniez commence à initier sa fille aux soins du ménage et cherche à lui inculquer des idées d'ordre- " l'ordre qu'est ce que c'est ? " demande Bécassine. " Il s'agit d'avoir une place pour chaque chose, de remettre toujours les choses à la même place, et de mettre autant que possible ,tous les objets pareils ensemble. - J'ons compris " fait Bécassine. Sa mère étant sortie, elle inspecte la salle et ouvre les meubles. Dans l'armoire, elle voit une pile de serviettes blanches tout près d'une pile de jupons en flanelle rouge " -du rouge et du blanc, c'est point pareils bien sûr!". Dans le buffet, même anomalie, voici un plat de fromage caillé à côté d'un saladier plein de tomates. Bécassine va arranger cela : elle ôte les jupons de l'armoire et met à sa place le caillé. Puis elle met les jupons dans le buffet, à voisiner avec les tomates : comme ça tout le blanc est ensemble et tout le rouge ensemble. C'est bien mieux. " Pinchon, l'enfance de Bécassine.*

### 5.1 Introduction

La classification automatique de données consiste à regrouper des objets, décrits par des attributs, en classes. Celles-ci contiennent les objets les plus proches selon un ou plusieurs critères et les objets éloignés devraient être dans des classes différentes.

La faculté de classier divers types d'objets dans leur milieu de vie, se retrouve chez plusieurs êtres vivants. Même si les mécanismes mis en jeu ne sont pas très bien compris, des méthodes dites bio-inspirées ont été proposées pour la classification de données. Par exemple, le comportement des fourmis, des abeilles et des mouches.

Les humains ont établi plusieurs classifications (le résultat d'une méthode de classification) dont, les hiérarchies du règne animale et végétale. Par exemple, la classification traditionnelle des êtres vivants.

Cependant, les humains rencontrent plusieurs difficultés liées au nombre d'attributs, au volume des données à traiter, ainsi qu'à l'indisponibilité d'informations a priori, comme une expertise sur le domaine de provenance des données.

Plusieurs approches ont été proposées pour résoudre ce problème et peuvent être rassemblées en trois grandes catégories : i) la classification supervisée ; ii) la classification non supervisée ou *clustering* et iii) la classification semi-supervisée.

La classification supervisée utilise les classes déjà connues pour déterminer les classes dans lesquelles les nouveaux objets seront placés.

La problématique de la classification non supervisée est la suivante : étant donné un certain nombre d'objets, sur lesquels peu ou prou d'informations sont disponibles, est-il possible d'identifier des classes dans ces données et s'il y en a le nombre exact ou un nombre de classes approprié pour les besoins de l'utilisateur.

La classification semi-supervisée est une approche intermédiaire entre les deux précédentes qui consiste à utiliser des informations a priori pour déterminer les classes. Un exemple est la méthode des k-moyennes, pour laquelle, le nombre de classes à déterminer est fixé à l'avance.

Ces trois approches peuvent être hybridées. Par exemple, le résultat d'une méthode semi-supervisée peut servir d'entrée à une méthode supervisée.

Dans ce chapitre, nous allons étudier la classification non supervisée.

## 5.2 Le processus du clustering

La clustering se déroule généralement selon le processus [Jain *et al.*, 1999] suivant :

- le prétraitement des données ;
- la définition d'une fonction de similarité appropriée aux données ;
- le clustering ;
- évaluation de la classification.

Nous allons détailler ces phases dans les sections suivantes.

### 5.2.1 Le prétraitement des données

Cette phase permet la détection et l'élimination des anomalies dans les données afin de fournir des données de qualité à l'algorithme de classification. Les anomalies se manifestent par des données :

- incomplètes : des valeurs sont manquantes pour certains attributs qui peuvent être importants eu égard au domaine d'application ;
- bruitées : contenant des erreurs, des valeurs aberrantes déviant des valeurs attendues ;
- inconsistantes : contenant des valeurs inappropriées pour décrire certains champs.

Des exemples de méthodes mises en œuvre pour ce faire sont :

- nettoyer les données : consiste à remplir les valeurs manquantes, lisser les données bruitées, identifier et supprimer les valeurs aberrantes et résoudre les inconsistances.
- intégration de données : consiste à fusionner des données provenant de plusieurs sources (des fichiers, des bases de données, des entrepôts de données) tout en évitant les redondances, les inconsistances.
- la transformation des données : consiste à normaliser les données afin de les ramener

$$\begin{bmatrix} x_{11} & \cdots & x_{1i} & \cdots & x_{1m} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{11} & \cdots & x_{i1} & \cdots & x_{im} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{ni} & \cdots & x_{nm} \end{bmatrix} \begin{bmatrix} 0 \\ d(2,1) & 0 \\ d(3,1) & d(3,2) & 0 \\ \vdots & \vdots & \vdots \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

TABLE 5.1 – Les deux structures de données utilisées pour représenter les relations entre les objets.

par exemple entre  $[0,0, 1,0]$  ;

- la réduction des données : consiste par exemple à sélectionner ou extraire les attributs qui décrivent mieux l'objet afin de réduire la dimension.

Néanmoins, certains choix, notamment, des transformations non appropriées, peuvent avoir des conséquences sur les autres phases [Han *et al.*, 2006].

## 5.2.2 Définitions

Soit  $X$  un ensemble de  $n$  éléments appelés objets. Un objet est en général décrit par  $m$  attributs prenant leurs valeurs dans un ensemble donné [Brucker et Barthélemy, 2007, Jain *et al.*, 1999]. D'autres synonymes comme données ou vecteurs de caractéristiques sont utilisés dans la littérature pour désigner un objet. Les attributs sont aussi nommés par caractéristiques.

Généralement, les objets sont exploités sous deux formes par les algorithmes : i) une matrice d'objets représentant  $n$  objets définie par  $m$  ; ii) une matrice de dissimilarité rassemblant les valeurs de la fonction de dissimilarité choisie pour chaque paire d'objets . Ces deux types de structures sont illustrées par la Figure 5.1.

La similarité est une mesure utilisée pour déterminer la proximité entre deux objets. Plus la similarité est grande plus les objets sont proches et moins elle est grande, plus les objets sont éloignés.

La dissimilarité, définie sur l'ensemble  $X$ , est une fonction  $d$  du produit cartésien  $X \times X$  dans  $R$  telle que [Brucker et Barthélemy, 2007] :

$$d(x, y) \geq 0 \text{ pour tout } x, y \in X, \quad (5.1)$$

$$d(x, y) = d(y, x) \text{ pour tout } x, y \in X, \quad (5.2)$$

$$d(x, x) = 0 \text{ pour tout } x \in X, \quad (5.3)$$

La distance est une dissimilarité qui vérifie l'inégalité triangulaire :

$$d(x, y) + d(y, z) \geq d(x, z) \text{ pour tout } x, y, z \in X \quad (5.4)$$

La définition d'une fonction de dissimilarité est cruciale pour la classification, car elle conditionne, dans certains cas, la forme des classes trouvées par l'algorithme. Ce choix dépend en général du domaine d'application.

Nous allons décrire les types d'attributs suivants : numérique, binaire, ordinaire, catégorielle, hétérogène.

### 5.2.2.1 Les attributs numériques

Un objet  $x_i \in X$  est codé par un  $m$ -uplet,  $(x_{i1}, \dots, x_{im})$  de nombres réels avec  $X = R^m$ . Ces données sont obtenues par exemple par mesure de grandeurs physiques. Ainsi l'unité de mesure utilisée peut influencer l'algorithme de classification si des conversions entre unités de mesure ne sont pas proprement effectuées. Il est de ce fait utile de procéder à une standardisation ou normalisation des données afin de donner à chaque attribut le même poids. Le choix d'utiliser ou non une méthode de standardisation ou de normalisation est laissé à l'expert du domaine [Han *et al.*, 2006].

La mesure de dissimilarité la plus utilisée pour les variables numériques est la distance de Minkowski ou  $p$ -distance. Cette distance est décrite par la formule suivante :

$$d(x_i, x_j) = \sqrt[p]{(x_{i1} - x_{j1})^p + (x_{i2} - x_{j2})^p + \dots + (x_{im} - x_{jm})^p} \quad (5.5)$$

Avec  $x_i, x_j \in X$ . Lorsque  $p = 1, 2$ , on obtient respectivement la distance de Manhattan et la distance euclidienne. La forme des classes trouvées par un algorithme en se basant sur la distance euclidienne est sphérique. Pour  $p = \infty$ , la distance est :

$$d_\infty(x_i, x_j) = \max_{1 \leq k \leq m} |x_{ik} - y_{jk}| \quad (5.6)$$

Ces trois distances sont les plus utilisées en analyse de données [Brucker et Barthélemy, 2007].

### 5.2.2.2 Les attributs binaires

Un attribut binaire est décrit par deux modalités qui sont par convention 0 et 1. L'espace de représentation  $X$  est alors égal à  $\{0, 1\}^m$  [Brucker et Barthélemy, 2007]. L'utilisation d'une  $p$ -distance sur ce type d'attributs pourrait conduire à de mauvais résultats [Han *et al.*, 2006]. De ce fait, nous allons définir d'autres mesures de dissimilarité. Soient  $x_i, x_j \in X$  deux objets alors on a :

- $q$  le nombre d'attributs valant 1 pour les deux objets  $x_i$  et  $x_j$  ;
- $r$  le nombre d'attributs valant 1 pour  $x_i$  mais valant 0 pour  $x_j$  ;
- $s$  le nombre d'attributs valant 0 pour  $x_i$  et 1 pour  $x_j$  ;
- $t$  le nombre d'attributs valant 0 pour  $x_i$  et  $x_j$ .

Le nombre total est  $p = q + r + s + t$ .

Deux cas de figure se présentent pour déterminer la dissimilarité : i) cas où les attributs sont symétriques (dichotomiques) ; ii) le cas où ils sont asymétriques (présence-absence).

Un attribut est symétrique lorsque les deux modalités ont la même importance. Un exemple peut être un attribut « sexe » codé par 1 ou 2. Dans ce cas, la fonction de dissimilarité est définie par :

$$d(x_i, x_j) = \frac{r + s}{q + r + s + t} \quad (5.7)$$

Un attribut est asymétrique lorsque les états n'ont pas la même importance, comme pour les tests de maladie. Par convention, le cas le plus important est codé par 1 (par exemple test positif pour la maladie considérée) et 0 (test négatif). La fonction de dissimilarité est définie par :

$$d(x_i, x_j) = \frac{r + s}{q + r + s} \quad (5.8)$$

L'absence du  $t$  dans cette fonction est justifiée par le fait que, pour des variables asymétriques,  $t$  est moins significatif que  $q$ .

A partir d'une fonction de dissimilarité, peut être définie une fonction de similarité, appelé aussi coefficient de Jaccard, qui, dans le cas des variables binaires asymétriques est [Han *et al.*, 2006] :

$$sim(x_i, x_j) = \frac{q}{q + r + s} = 1 - d(x_i, x_j) \quad (5.9)$$

Il arrive souvent pour un même objet que des attributs des deux types de variables binaires soient présents. Dans ce cas, la fonction de dissimilarité est définie comme dans le cas des variables hétérogènes (voir la section 5.2.2.5). Il existe d'autres mesures de dissimilarité qui sont données par [Cornuéjols et Miclet, 2010] :

Nom	Formule	Remarque
Hamming	$\frac{s+t}{q+r+s+t}$	Est une distance
Russel et Rao	$1 - \frac{q}{q+r+s+t}$	Ne vérifie pas la première propriété de la distance
Yule	$1 - \frac{qt-rs}{qs-rt}$	

### 5.2.2.3 Les attributs catégoriels

Un attribut catégoriel est une généralisation d'un attribut binaire et ayant plus que deux états. Soit  $M$ , le nombre d'états d'une variable catégorielle, les états peuvent prendre des valeurs de  $1, \dots, M$ . Ici, l'ordre des états n'a pas d'importance.

La dissimilarité entre deux objets  $x_i$  et  $x_j$  est déterminée par le ratio :

$$d(x_i, x_j) = \frac{p - m}{p} \quad (5.10)$$

où  $m$  est le nombre d'attributs pour lesquels  $x_i$  et  $x_j$  ont le même état et  $p$  est le nombre total d'attributs.

Un attribut catégoriel peut être recodé en attributs binaires asymétriques en créant un attribut pour chacun des  $M$  états [Han *et al.*, 2006].

### 5.2.2.4 Les attributs ordinaux

Un attribut ordinal peut être de deux types : discret ou continu. Un attribut ordinal est comme un attribut catégoriel dont l'ordre des  $M$  états est important. Ces variables sont utiles pour coder des valeurs subjectives comme les catégories socioprofessionnelles [Han *et al.*, 2006].

La définition d'une fonction de dissimilarité pour les attributs ordinaux est la suivante : Considérant un attribut  $f$  appartenant à un ensemble d'attributs ordinaux décrivant  $n$  objets, la dissimilarité par rapport à  $f$  est déterminée suivant les étapes [Han *et al.*, 2006] :

1. La valeur de  $f$  pour le  $i^{\text{ème}}$  objet est  $x_{if}$ , et  $f$  a une valeur  $M_f$  dans l'ordre des états. Remplacer chaque  $x_{if}$  par son rang  $r_{if} \in \{1, \dots, M_f\}$ .
2. Les valeurs des ces rangs sont ramenées entre  $[0.0, 1.0]$  afin d'accorder à chacun le même poids. Chaque rang est ainsi remplacé par :

$$z_{if} = \frac{r_{if} - 1}{M_f - 1} \quad (5.11)$$

3. La dissimilarité peut être déterminée par les distances décrites pour les variables numériques.

### 5.2.2.5 Les objets à attributs hétérogènes

Cette catégorie d'objets est celle qu'on rencontre dans les bases de données réelles. Avec l'accroissement des capacités de stockage et l'apparition de nouveaux domaines d'application de la classification (les réseaux sociaux, les réseaux de capteurs, etc.), d'autres types de difficultés sont à prendre en compte à savoir :

- les bases de données de grand volume, non structurées (nombre d'attributs différents d'un objet à l'autre), les données sont de types variés (multimédia) [Jain, 2010];
- la base de données distribuées sur plusieurs sites;

Nous allons donner un aperçu sur trois catégories [Jain, 2010] de données : i) dynamiques, ii) représentées par des graphes et iii) relationnelles.

1. Les données dynamiques sont des données qui changent au cours du temps par opposition aux données statiques. Ce sont par exemple les pages Web, les séries temporelles et les flux de paquets dans les réseaux. Le problème consiste à classifier ces données volumineuses, de taille potentiellement non limitée et reçues de façon séquentielle par l'algorithme. Ces données exigent des algorithmes de :
  - la rapidité de traitement de quantités massives de données arrivant au fur et à mesure;
  - détecter les classes émergentes et les séparer des valeurs aberrantes dans les données;
  - accorder une durée de vie aux classes trouvées (une classe valide à l'instant  $t$  ne l'est plus à  $t + 1$  par exemple) [Jain, 2010].

Les algorithmes proposés pour ces données sont en général des adaptations des méthodes déjà existantes [Cao *et al.*, 2006, Hore *et al.*, 2009] : *k*-means, *fuzzy c-means*, etc.

2. Les données représentées par des graphes sont traitées de plusieurs façons : i) transformer le graphe représentant l'objet en vecteur de caractéristiques et appliquer des méthodes de partitionnement de ces graphes en groupes distincts ; ii) développer des fonctions noyau plus adaptées aux données à base de graphes. La similarité peut être déterminée par la méthode décrite par [Kashima *et al.*, 2003]; iii) aligner les matrices d'adjacence des graphes [Umeyama, 1988].
3. Les données relationnelles codent des relations entre des entités. La classification de données relationnelles consiste à partitionner un large graphe (un réseau) en des sous-graphes cohésifs basés sur la structure des liens et des attributs des nœuds.

### 5.3. LE PROBLÈME ÉTUDIÉ

---

Le problème devient plus compliqué lorsque les liens (qui représentent des relations entre objets) sont autorisée sous plusieurs types. Un choix critique est de définir le critère approprié sur lequel la classification sera basée. Un certain nombre de modèles ont été proposés dont :

- un modèle probabiliste général dans [Taskar *et al.*, 2001] ;
- la fonction de modularité de Newman (*Newman Modularity Function*) et une relaxation spectrale de la fonction [Newman et Girvan, 2004] de Newton et Girvan

Deux approches peuvent être utilisées en ce qui concerne le calcul de la dissimilarité entre des objets hétérogènes.

La première consiste à classer les objets par rapport à chaque type d'attribut. Si les objets ont  $p$  attributs, nous pouvons utiliser  $p$  algorithmes différents. Seulement cette approche ne donne pas de bons résultats dans les applications réelles.

La seconde approche consiste à utiliser un seul algorithme en ramenant les valeurs des divers attributs entre  $[0.0, 1.0]$  et déterminer une matrice de dissimilarité. Soient deux objets  $x_i$  et  $x_j$ , la dissimilarité entre ces deux objets est donnée par [Han *et al.*, 2006] :

$$d(i, j) = \frac{\sum_{f=1}^p \delta_{ij}^{(f)} d_{ij}}{\sum_{f=1}^p \delta_{ij}^{(f)}} \quad (5.12)$$

où le nombre de Kronecker  $\delta_{ij}^{(f)} = 0$  si : i) les valeurs de  $x_{if}$  et  $x_{jf}$  sont manquantes pour les objets  $x_i$  et  $x_j$ , ii)  $x_{if} = x_{jf} = 0$  et  $f$  est binaire asymétrique. Sinon  $\delta_{ij}^{(f)} = 1$ . La dissimilarité entre les deux objets est alors déterminée par  $\delta_{ij}^{(f)}$  qui est :

- si  $f$  est numérique  $\delta_{ij}^{(f)} = \frac{|x_{if} - x_{jf}|}{\max_h x_{hf} - \min_h x_{hf}}$ , où  $h = 1, \dots, n$  pour la variable  $f$ .
- si  $f$  est binaire ou catégorique :  $\delta_{ij}^{(f)} = 0$  si  $x_{if} = x_{jf}$  ; sinon  $\delta_{ij}^{(f)} = 1$
- si  $f$  est ordinale, appliquer la méthode décrite dans la sous-section 5.2.2.4.

Les étapes décrites sont celles que nous avons déjà rencontrées pour des objets décrits par un seul type. Sauf pour les variables numériques qui sont normalisées entre  $[0.0, 1.0]$ .

D'autres mesures de dissimilarité sont utilisées pour des objets exprimés comme des vecteurs, ceci est le cas pour la recherche d'informations sur internet ou la classification de documents textuels. Ces fonctions sont capables de mieux gérer des objets complexes comme les documents contenant un grand nombre d'objets symboliques (mots-clés, hyperliens, phrases, etc). Les mesures de la similarité sont dans ce cas des fonctions *non-métriques*. Il en existe plusieurs comme la fonction cosinus.

### 5.3 Le problème étudié

Nous nous sommes intéressés à l'étude de la classification par algorithmes de fourmis, qui sont une branche des algorithmes biomimétiques, afin de mieux comprendre leur fonctionnement, leurs contributions et limitations. L'introduction d'une nouvelle méthode doit, à notre avis, s'accompagner d'une étude approfondie de ses apports et limitations par rapport aux méthodes déjà existantes et apparentées. En utilisant des bases de données autant artificielles que des données provenant de la base de *Machine Learning* et d'applications

réelles. Il est aussi important d'étudier les fonctions d'évaluation les plus couramment utilisées pour évaluer des algorithmes de fourmis. Ceci parce que durant l'évaluation d'un algorithme, les choix de la base de données et de des fonctions d'évaluation ne sont pas neutres. Il en est de même des algorithmes contre lesquels il faut comparer ses résultats. Nous voulons donner ici au lecteur un large aperçu sur les autres méthodes de classification afin qu'il situe mieux dans quel contexte s'inscrit le travail présenté dans cette deuxième partie de la thèse ainsi que notre contribution à la classification par fourmis artificielles.

Nous avons aussi établi que plusieurs algorithmes de fourmis proposés pour la classification de données s'inspirent des paradigmes suivants : les méthodes de partitionnement, les méthodes hiérarchiques, les méthodes à base de densité et les méthodes probabilistes. Ces paradigmes et les hypothèses sous-jacentes sont décrites dans les sections suivantes.

Les sections suivantes sont consacrées à la description de ces méthodes.

## 5.4 Les méthodes de partitionnement

Une méthode de partitionnement détermine un nombre  $k$  de classes pour un ensemble de données, où  $k \leq n$  ( $n$  est le nombre d'objets). Il n'est pas toujours facile de définir la notion de classe. Une fois que cette définition est établie, elle conditionne la méthode à utiliser. Pour les méthodes de partitionnement, l'hypothèse est qu'il existe au moins  $k$  classes dans les données qui sont des sous-ensembles non vides de  $X$  ( $X$  est l'ensemble des données). Mais en général, pour un ensemble de données, on peut former  $2^n - 1$  classes.

Ces classes peuvent se différencier selon leur forme, taille et densité [Jain, 2010] et satisfont aux contraintes suivantes [Han *et al.*, 2006] :

- chaque classe doit contenir au minimum un objet ;
- chaque objet doit appartenir à une seule classe. Cette contrainte peut être relâchée pour certains algorithmes dits flous (*fuzzy*).

Une classe idéale peut être définie comme un ensemble de points « compacts » et « isolés » [Jain, 2010].

Des critères permettent de mesurer combien les classes sont compactes et isolées. En d'autres termes, mesurer la qualité d'un partitionnement consiste à déterminer si les objets d'une même classe sont proches les uns des autres ou reliés entre eux tandis qu'ils sont loin des autres objets et ne sont pas liés à eux.

Deux grandes catégories de critères peuvent être établies [Brucker et Barthélemy, 2007] :

1. les critères d'homogénéité qui indiquent en quoi les éléments d'une même classe se ressemblent ;
2. les critères de séparabilité qui indiquent en quoi les éléments d'une classe diffèrent des éléments des autres classes.

Les méthodes de partitionnement commencent par créer une partition initiale de  $k$  classes et utilisent une méthode itérative d'allocation-recentrage pour améliorer le partitionnement en transférant les objets d'une classe à l'autre.

Une partition est définie par [Brucker et Barthélemy, 2007] :

**Définition 5.4.1** Une partition  $P = \{C_1, \dots, C_k\}$  sur  $X$  est une collection de  $X$  telle

que :

- l'union des éléments de  $P$  soit égale à  $X$  ( $\cup_{1 \leq i \leq k} C_i = X$ )
- pour tout  $1 \leq i \neq j \leq k$   $C_i \cap C_j = \emptyset$

Une collection étant un ensemble de sous-ensembles non vides de  $X$ .

**Définition 5.4.2** On désigne par  $\Pi$  l'ensemble de toutes les partitions de  $X$ . Soient deux partitions  $\pi_i$  et  $\pi_j$ .  $\pi_i$  est plus fine que  $\pi_j$  si et seulement si toute classe de  $\pi_j$  est une classe de  $\pi_i$  ou l'union de plusieurs classes de  $\pi_i$ .

**Définition 5.4.3** Une chaîne [Cornuéjols et Miclet, 2010] dans l'ensemble des partitions de  $X$  est un ensemble de partitions  $\{\pi_1, \dots, \pi_r\}$  tel que pour  $i = 1, r - 1$  on a :  $\pi_i$  est plus fine  $\pi_{i+1}$ .

Le nombre de partitions en  $k$  classes de l'ensemble  $X$  s'appelle couramment *nombre de Stirling de seconde espèce* et se note  $S(n, k)$  :

$$S(n, k) = \frac{1}{k} \sum_{i=0}^k (-1)^i C_k^i (k-i)^n \quad (5.13)$$

Ce nombre croit exponentiellement même pour des petites bases de données. Des heuristiques ont été proposées pour trouver un partitionnement avec un compromis qualité/temps. Nous allons présenter deux approches pour évaluer la qualité d'un algorithme de partitionnement : évaluer et comparer les classes d'une partition et évaluer et comparer des partitions entre elles.

### 5.4.1 Évaluation et Comparaison de classes

Nous allons utiliser les critères suivants pour évaluer la classe  $c_i$  d'une partition  $P$ . Soient : une dissimilarité  $d$ , une classe  $c$ , les cardinalités de  $|X|$  et de  $|c_i|$ ,

- le diamètre de  $c_i$  est la dissimilarité entre les objets de  $c_i$  les plus éloignés.

$$diam(c_i) = \max\{d(x, y) \mid (x, y) \in c_i\} \quad (5.14)$$

$diam(c_i)$  est lié à l'homogénéité de la classe, plus le diamètre est petit, plus la classe est homogène.

- le rayon est calculé par rapport à un point (centre de gravité) de la classe dont les autres éléments de  $c_i$  doivent être proches. Il est déterminé par la formule :

$$r(c_i) = \min\{\max\{d(x, y) \mid y \in c_i - \{x\}\} \mid x \in c_i\} \quad (5.15)$$

Le rayon est aussi lié à l'homogénéité d'une classe. Il est à minimiser.

- la séparation de  $c_i$  mesure la dissimilarité minimale entre les objets les plus proches, l'un étant dans  $c_i$ , l'autre dans d'autres classes :

$$sep(c_i) = \min\{\max\{d(x, y) \mid x \in c_i\} \mid y \notin c_i\} \quad (5.16)$$

Plus la séparation est grande, plus les objets de  $c_i$  sont isolés des autres objets.

- l'étoilement de  $c_i$  détermine la valeur minimale de la somme des dissimilarités entre un objet de  $c_i$  et les autres objets de  $c_i$

$$et(c_i) = \min \left\{ \sum_{y \in c_i} d(x, y) \mid x \in c_i \right\} \quad (5.17)$$

Il est lié à l'homogénéité et est la somme des rayons en considérant, à tour de rôle, chaque objet comme centre de la classe.

- l'indice de clique de  $c_i$  est la somme des dissimilarités, des diamètres, entre les éléments de  $c_i$  :

$$cl(c_i) = \sum_{x, y \in c_i} d(x, y) \quad (5.18)$$

Il est lié à l'homogénéité.

Pour les variables numériques où un centre de gravité,  $g$ , peut être calculé, nous allons considérer l'inertie et la variance.

- L'inertie  $I(c_i)$  d'une classe est définie par :

$$I(c_i) = \frac{1}{|X|} \sum_{x \in c_i} \|x - g\|^2 \quad (5.19)$$

- et la variance  $V(c_i)$  de la classe est donnée par :

$$V(c_i) = \frac{1}{|c_i|} \sum_{x \in c_i} \|x - g\|^2 \quad (5.20)$$

L'inertie et la variance mesurent l'homogénéité d'une classe, plus elles sont petites, plus les objets de la classe sont compacts autour du centre de gravité. Une bonne classe est une classe sphérique de petit rayon [Brucker et Barthélemy, 2007]. Ces deux indices peuvent aussi être généralisés aux cas d'une fonction de dissimilarité quelconque et de ce fait prendre en compte plusieurs types d'attributs. Ils deviennent :

$$I(c_i) = \frac{1}{2|X||c_i|} \sum_{x, y \in c_i} d(x, y)^2 \quad (5.21)$$

$$V(c_i) = \frac{1}{2|c_i|^2} \sum_{x, y \in c_i} d(x, y)^2 \quad (5.22)$$

Pour comparer des classes entre elles afin de savoir si elles sont proches ou éloignées, l'espace des classes est muni d'une dissimilarité appelée couramment *mesure de ressemblances*. Soient  $c_i$  et  $c_j$  deux classes de  $P$ , quelques mesures de ressemblance entre ces deux

classes sont données par [Brucker et Barthélemy, 2007] :

$$1 - \frac{|c_i \cap c_j|}{|c_i \cup c_j|}, \text{ (Jaccard, 1901)} \quad (5.23)$$

$$1 - \frac{|c_i \cap c_j|}{2|c_i \cup c_j| - |c_i \cap c_j|}, \text{ (Sokal et Sneath (1963), Anderberg (1973))} \quad (5.24)$$

$$1 - \frac{|c_i \cap c_j|}{\sqrt{|c_i| \cdot |c_j|}}, \text{ (Ochiai (1957), Driver et Kroeber (1932))} \quad (5.25)$$

$$1 - \frac{2|c_i \cap c_j|}{|c_i| + |c_j|}, \text{ (Czeknowski (1913), Dice(1945))} \quad (5.26)$$

$$1 - \frac{|c_i \cap c_j|}{\max |c_i|, |c_j|}, \text{ (Braun-blanquet (1932))} \quad (5.27)$$

$$1 - \frac{|c_i \cap c_j|}{\min |c_i|, |c_j|}, \text{ (Simpson (1943)).} \quad (5.28)$$

### 5.4.2 Évaluation d'une partition

La mesure de la qualité d'une partition peut être obtenue par généralisation des indices présentés pour une classe. Soit  $P = \{c_1, \dots, c_k\}$  une partition sur  $X$ .

On définit [Brucker et Barthélemy, 2007] :

— la somme des diamètres :

$$f_{\sum diam}(P) = \sum_{1 \leq i \leq k} diam(c_i) \quad (5.29)$$

— le diamètre maximum :

$$f_{\max diam}(P) = \max\{diam(c_i) | 1 \leq i \leq k\} \quad (5.30)$$

— la somme des indices de clique :

$$f_{\max cl}(P) = \sum \{cl(c_i) | 1 \leq i \leq k\} \quad (5.31)$$

— la somme de l'étoilement :

$$f_{\sum et}(P) = \sum_{1 \leq i \leq k} et(c_i) \quad (5.32)$$

— le maximum du rayon :

$$f_{\max rd}(P) = \max\{r(c_i) | 1 \leq i \leq k\} \quad (5.33)$$

Les indices de classe choisis, tous liés à un critère d'homogénéité, montrent qu'une partition sera d'autant meilleure que son critère de qualité est petit. Une fois un critère choisi, la meilleure partition sera donc celle réalisant le minimum de celui-ci.

Le problème de classification peut ainsi être redéfini comme un problème d'optimisation. Des algorithmes construisant la partition optimale pour chacun des critères sont donnés

dans [Brucker et Barthélemy, 2007]. Dans ce cas, le minimum est toujours atteint pour la partition ne contenant que les classes *singletons*. Ce problème est évité par la fixation du nombre de classes à trouver qui optimisent le critère [Brucker et Barthélemy, 2007].

En considérant une distance euclidienne, la mesure de qualité la plus utilisée est la somme de l'inertie de chaque classe de la partition. Pour une partition  $P = \{c_1, \dots, c_k\}$  on a :

$$f_I(P) = \sum_{1 \leq i \leq k} I(c_i) \quad (5.34)$$

Elle s'appelle aussi *inertie intra-classe*. Bien que cet indice soit lié à l'homogénéité des classes, il s'interprète également comme un critère lié à la séparation. On peut en effet définir l'indice de l'inertie inter-classe en déterminant la norme  $L_2^1$  entre les centres de gravité.

$$f_{I_B}(P) = \sum_{1 \leq i \leq k} \frac{|c_i|}{|X|} \|g(X) - g(c_i)\|_2^2 \quad (5.35)$$

Où  $g(c_i)$  est le centre de gravité. Plus l'inertie inter-classe est élevée, plus les classes sont séparées les unes des autres : la meilleure partition selon ce critère est celle qui la maximise [Brucker et Barthélemy, 2007].

L'indice de Dunn [Dunn, 1973] est basé sur l'idée d'identifier un partitionnement qui est compact et dont les classes sont bien séparées. En considérant un partitionnement  $P = \{c_1, \dots, c_i, \dots, c_n\}$ , l'index de Dunn  $D$  pour la partition  $P$  est :

$$D = \min_{1 \leq i \leq n} \left\{ \min_{1 \leq j \leq n, j \neq i} \left\{ \frac{\delta(c_i, c_j)}{\max_{1 \leq k \leq n} \Delta_k} \right\} \right\} \vee i, j, k \quad (5.36)$$

Où :  $\delta(c_i, c_j)$  représente la distance entre les classes  $c_i$  et  $c_j$  et  $\Delta_k$  est le diamètre de la classe  $k$  qui peut être déterminé de plusieurs façon. L'indice de Dunn est à maximiser.

L'indice de Davies-Bouldin est une fonction du ratio de la somme des étoilements des classes sur la séparation entre les classes. Il est défini par :

$$DB = \frac{1}{n} \sum_{i=1}^n \max_{i \neq j} \left\{ \frac{\bar{d}(c_i) + \bar{d}(c_j)}{\delta(c_i, c_j)} \right\} \quad (5.37)$$

Où  $\bar{d}(c_i)$  est la distance moyenne de tous les objets de la classes  $c_i$  à leur centre de gravité,  $\delta(c_i, c_j)$  est la distance entre les centres de gravité. La meilleure partition en utilisant cet indice est celle qui la minimise.

### 5.4.3 Comparaison de partitions

La comparaison de plusieurs partitions d'un même ensemble de données peut se faire pour mesurer leurs différences et aussi déterminer la meilleure pour un domaine d'applica-

---

1. La norme  $L_2$  ou norme euclidienne est définie pour  $x$  par  $\|x\|_2 = \sqrt{\sum_{1 \leq k \leq m} |x^k|^2}$ . Cette norme peut être généralisée en  $L_p$  en remplaçant le 2 par  $p$ .

tion. Les performances d'un algorithme peuvent être mesurées en comparant le partitionnement obtenu avec celui dit « naturel », ce qui est généralement possible avec les bases de données de *benchmarking*.

Comme cela a été rapporté par [F. et Rohlf, 1982, Hubert et Arabie, 1985], la comparaison de partitions peut se faire selon plusieurs approches dont la plus populaire est celle dite des matrices de contingences.

Une autre approche consiste à mesurer le nombre de transformations élémentaires permettant de passer d'une partition à l'autre [Brucker et Barthélemy, 2007]. Ainsi sur la base de ces transformations, des distances peuvent être définies, par exemple la distance de transfert [Régnier, 1965] et la distance de Boorman [Boorman et Arabie, 1972].

Une matrice de contingence contient les informations sur la répartition des objets dans les classes des deux partitions. Soit  $P = \{p_1, p_2, \dots, p_M\}$  et  $Q = \{q_1, q_2, \dots, q_N\}$  deux partitions sur  $X$  de nombre de classes respectifs  $M$  et  $N$ , la matrice de contingence pour  $P$  et  $Q$  est donnée par [Hubert et Arabie, 1985] :

Classe	$p_1$	$\dots$	$p_j$	$\dots$	$p_M$	Sommes
$q_1$	$n_{11}$	$\dots$	$n_{1j}$	$\dots$	$n_{1M}$	$n_{1.}$
$\vdots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
$q_i$	$n_{i1}$	$\dots$	$n_{ij}$	$\dots$	$n_{iM}$	$n_{i.}$
$\vdots$						
$q_N$	$n_{N1}$	$\dots$	$n_{Nj}$	$\dots$	$n_{NM}$	$n_{N.}$
Sommes	$n_{.1}$	$\dots$	$n_{.j}$	$\dots$	$n_{.M}$	$n_{..} = n$

Où :

- $n_{ij}$  le nombre d'objets appartenant à l'intersection des classes  $p_i$  et  $q_j$ .
- $n_{.i}$  le nombre d'objets dans la classe  $p_i$  ;
- $n_{.j}$  le nombre d'objets dans la classe  $q_j$  ;
- $n$  le nombre d'objets de la base de données.

Les indices de performance utilisant la matrice de contingence sont basés sur le classement d'une paire d'objets dans les deux partitions. Quatre possibilités existent et pour chacune, le nombre de paires est :

1. les objets de la paire sont classés dans la même classe dans la partition  $P$  et dans la partition  $Q$ .

$$\frac{1}{2} \sum_{i=1}^M \sum_{j=1}^N (n_{ij} (n_{ij} - 1)) \quad (5.38)$$

2. les objets de la paire sont classés dans des classes différentes dans  $P$  et dans des classes différentes dans  $Q$  ;

$$\frac{1}{2} \left( n^2 + \sum_{i=1}^M \sum_{j=1}^N n_{ij}^2 - \left( \sum_{i=1}^M n_{i.}^2 + \sum_{j=1}^N n_{.j}^2 \right) \right) \quad (5.39)$$

3. les objets de la paires sont classés dans des classes différentes dans la partition  $P$

mais dans la même classe pour la partition  $Q$  ;

$$\frac{1}{2} \left( \sum_{j=1}^N n_{.j}^2 - \sum_{i=1}^M \sum_{j=1}^N n_{ij}^2 \right) \quad (5.40)$$

4. les objets de la paires sont classés dans la même classe dans la partition  $P$  mais dans des classes différentes pour la partition  $Q$ .

$$\frac{1}{2} \left( \sum_{i=1}^M n_i^2 - \sum_{i=1}^M \sum_{j=1}^N n_{ij}^2 \right) \quad (5.41)$$

Les types 1 et 2 représentent l'agrément  $A$  entre les deux partitions et le types 3 et 4 le désaccord  $D$ . Avec :

$$A = C_2^n + \sum_{i=1}^M \sum_{j=1}^N n_{ij}^2 - \frac{1}{2} \left( \sum_{i=1}^M n_i^2 + \sum_{j=1}^N n_{.j}^2 \right) \quad (5.42)$$

$$D = \frac{1}{2} \left( \sum_{i=1}^M n_i^2 + \sum_{j=1}^N n_{.j}^2 \right) - \sum_{i=1}^M \sum_{j=1}^N n_{ij}^2 \quad (5.43)$$

La somme de  $A + D = C_2^n$  qui est le nombre total de combinaisons de paires que l'on peut construire sur  $X$ .

Ainsi intuitivement, deux partitions sont similaires lorsque la valeur de  $A$  est grande et celle de  $D$  est petite. Les indices basés sur ces deux mesures, diffèrent les uns des autres par la façon dont  $A$  et  $D$  sont normalisés. Ces indices sont [Hubert et Arabie, 1985] :

- l'indice de Rand se détermine comme :  $A/C_2^n$ . Cet indice est le plus populaire et a été beaucoup adapté. En considérant les paires d'objets tirées aléatoirement, l'indice de Rand peut être interprété comme la probabilité d'agrément ;
- l'indice dit de Johnson (aussi dit de Mirkin ou de Arabie et Boorman) se calcule comme :  $D/C_2^n$  et s'interprète comme la probabilité de désaccord ;
- l'indice de Wallace (1983) est basé sur deux mesures :

$$m_1 = \sum_{i,j} C_2^{n_{ij}} / \sum_i C_2^{n_i} \quad \text{et} \quad m_2 = \sum_{i,j} C_2^{n_{ij}} / \sum_j C_2^{n_{.j}} \quad (5.44)$$

$m_1$  et  $m_2$  ne prennent en compte que les paires du type 1 car Wallace considère que les paires du type 2 sont « neutres » et ne doivent pas être prises en charge dans le calcul de l'indice de correspondance. En considérant une partition  $P$  comme la partition naturelle contre laquelle il faut comparer  $Q$ , alors la première mesure peut être considérée comme la probabilité qu'une paire d'objets tirée aléatoirement soit dans la même classe dans la partition  $Q$  sachant qu'elle est dans la même classe dans la partition  $P$ . La seconde mesure s'interprète de la même manière, en considérant  $Q$  comme partition naturelle. L'indice de Wallace est alors la moyenne géométrique de ces deux mesures :

$$\sum_{i,j} C_2^{n_{ij}} / \sqrt{m_1 \times m_2} \quad (5.45)$$

Hubert et Arabie (1985) ont proposé une approche pour corriger ces indices. En se mettant sous l'hypothèse d'une distribution de probabilité géométrique, c'est-à-dire que les partitions sont considérées formées aléatoirement et, de ce fait peuvent, contenir les classes d'origine avec leurs nombres d'objets. Ainsi, une espérance  $E[Indice]$  peut être déterminée pour les objets du type 1. La correction proposée est donnée par la formule [Hubert et Arabie, 1985] :

$$Indice_{Ajust} = \frac{Indice - E[Indice]}{Maximum[Indice] - E[Indice]} \quad (5.46)$$

Pour l'indice de Rand, la correction donne :

$$\frac{\sum_{i,j} C_2^{n_{ij}} - \sum_i C_2^{n_i} \cdot \sum_j C_2^{n_j} / C_2^n}{\frac{1}{2}[\sum_i C_2^{n_i} + \sum_j C_2^{n_j}] - \sum_i C_2^{n_i} \cdot \sum_j C_2^{n_j} / C_2^n} \quad (5.47)$$

Avec pour valeur maximale 1 pour l'indice ajusté. Le choix d'une borne supérieure ou inférieure (dans le cas de l'indice Johnson) a conduit à une multitudes d'autres indices ajustés [Hubert et Arabie, 1985].

Trouver la partition qui optimise ces indices est un problème difficile. Des heuristiques permettent de trouver des compromis et dans certains cas convergent vers des optimums locaux. C'est le cas des k-means que nous allons développer.

#### 5.4.4 k-means : un exemple d'algorithme de partitionnement

L'algorithme k-means est l'un des algorithmes de partitionnement les plus populaires.

L'idée de base est de construire un partitionnement d'un ensemble de données en  $k$  classes qui minimise l'inertie intra-classe. Il est décrit par :

---

**Algorithme 5** : L'algorithme k-means

---

Choisir aléatoirement dans  $X$ ,  $k$  objets qui seront les centres des classes;

**Répéter**

1. (re)assigner chaque objet à la classe dont il est le plus proche en se basant sur la distance au centre de la classe;
2. recalculer les centres (moyennes de tous les objets dans la classe) des classes;

**Jusqu'à** ce qu'il n'y ait aucun changement;

---

L'algorithme k-means fonctionne mieux lorsque les classes forment des nuages compacts de points et sont bien séparés les uns des autres. La méthode n'est pas adaptée pour trouver des classes de forme non-convexe et des classes de différentes tailles. Elle est aussi sensible aux bruits et aux valeurs aberrantes qui peuvent influencer la moyenne. Assez souvent la méthode termine sur un optimum local et ne peut être appliquée que dans les cas où la moyenne peut être définie, ce qui n'est pas le cas pour certains types de données (ex. les données catégorielles). Aussi, le fait de spécifier la valeur de  $k$  peut être un inconvénient. Il existe un certain nombre de variantes de k-means qui diffèrent par leur façon de choisir les premiers centres des classes, de déterminer la dissimilarité, et la stratégie pour recalculer

les centres des classes. Une autre approche intéressante qui peut conduire à de bon résultats est l'hybridation avec d'autres méthodes de classification.  $k$ -means est alors appliqué soit en amont ou en aval de la méthode pour raffiner les résultats. Voici une liste non exhaustive des variantes de  $k$ -means : ISODATA [Ball et Hall, 1965], FORGY [Forgy, 1965], Fuzzy C-means [Dunn, 1973, Bezdek, 1981],  $x$ -means [Pelleg et Moore, 2000], kernel  $k$ -means [Schölkopf *et al.*, 1998, Dhillon *et al.*, 2004].

On peut aussi remplacer la moyenne par les autres mesures de tendance centrale : mode, médiane pour rendre la méthode moins sensible aux valeurs aberrantes et plus appropriée pour les données catégoriques.

Ces modifications introduisent néanmoins de nouveaux paramètres à spécifier dont les choix ne sont pas évidents.

#### 5.4.5 La détermination du nombre de classes $k$

L'un des inconvénients de la méthode  $k$ -means et, en général, des méthodes de partitionnement est que l'on doit fixer le nombre de classes,  $k$ , à l'avance. Plusieurs méthodes ont été mises en œuvre pour déterminer automatiquement  $k$  en fonction des données.

Une première approche consiste à exécuter l'algorithme avec des valeurs différentes pour  $k$  et de choisir celle qui optimise un critère.

Une méthode donnée par [Figueiredo et Jain, 2002] utilise le critère MML (Minimum Message Length [Wallace et Boulton, 1968]) en conjonction avec un mélange de gaussiennes pour estimer  $k$ . L'approche commence par un grand nombre de classes et fusionne au fur et à mesure des classes si cela contribue à minimiser le critère MML. Cette approche peut être généralisée en minimisant d'autres critères [Jain, 2010] :

- Minimum Description Length [Hansen et Yu, 2001] ;
- Bayesian Information Criterion [Schwarz, 1978] ;
- Akaike Information Criterion [Sakamoto *et al.*, 1986].

Gap statistics [Tibshirani *et al.*, 2000] et le processus de Dirichlet (DP) [Ferguson, 1973, Rasmussen, 2000] sont aussi utilisés pour estimer  $k$ .

En dépit de ces méthodes, l'estimation du nombre de classes reste toujours une tâche difficile [Jain, 2010].

## 5.5 Les méthodes hiérarchiques

Les méthodes de partitionnement souffrent du fait que le nombre de classes doit être fourni et aussi que les objets sont assignés à une et une seule classe (cette contrainte est relâchée pour certaines méthodes). L'approche adoptée par les méthodes hiérarchiques consiste à agglomérer les objets, ou à diviser la base de données, jusqu'à regrouper tous les objets dans la même classe, ou chaque objet dans une seule classe. Ainsi, selon le moment où le processus est arrêté, la méthode retourne un nombre de classes. Ces deux approches sont illustrées sur la Figure 5.1 par deux algorithmes AGNES (Agglomerative Nesting) et DIANA (Divisive Analysis) appliqués sur un ensemble  $X = \{a, b, c, d, e\}$ .

Une méthode hiérarchique crée des hiérarchies sur l'ensemble  $X$ . Une hiérarchie est

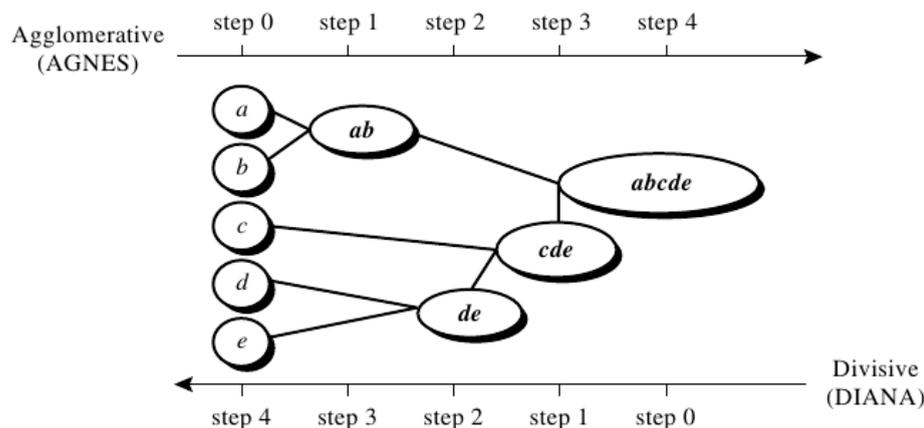


FIGURE 5.1 – Les algorithmes AGNES et DIANA appliquant les deux approches : agglomérative et divisive [Handl et Meyer, 2007].

définie par [Cornuéjols et Miclet, 2010] :

**Définition 5.5.1** Une hiérarchie  $H$  sur l'ensemble des données  $X$  est un sous-ensemble des parties de  $X$  tel que :

- pour tout élément  $x$  de  $X$ ,  $\{x\} \in H$  ;
- pour tout couple d'éléments  $h$  et  $h'$  de  $H$  avec  $h \neq h'$ , on a :
  - Soit  $h \cap h' = \emptyset$ ,
  - Soit  $h \cap h' \neq \emptyset$ , alors  $h \subset h'$ , soit  $h' \subset h$

**Définition 5.5.2** Une hiérarchie indicée est une hiérarchie  $H$  sur un ensemble fini à laquelle on associe une suite de nombres réels  $r_i$ . Une hiérarchie indicée est monotone si pour deux éléments  $h_i$  et  $h_{i+1}$  consécutifs dans  $H$ , avec  $h_i$  plus fine que  $h_{i+1}$ , on a  $r_i \leq r_{i+1}$ .

Soit un ensemble  $X = \{a, b, c, d, e, f\}$ . Un exemple de hiérarchie sur  $X$  est donné par :  $H = \{h_1, \dots, h_{10}\}$  où :  $h_1 = \{a\}$ ,  $h_2 = \{b\}$ ,  $h_3 = \{c\}$ ,  $h_4 = \{d\}$ ,  $h_5 = \{e\}$ ,  $h_6 = \{f\}$ ,  $h_7 = \{a, b, c, d, e, f\}$ ,  $h_8 = \{a, b, c, d\}$ ,  $h_9 = \{e, f\}$ ,  $h_{10} = \{a, b, c\}$

Une hiérarchie est généralement représentée sous forme d'un dendrogramme qui est illustré pour  $H$  par la Figure 5.2. Les résultats montrés par cette figure sont les résultats d'une étude présentés par [Jain *et al.*, 2004]. Cette étude avait pour but de classifier des algorithmes de classification en se basant sur 35 indices différents.

La hiérarchie indicée est représentée avec les valeurs des indices pour chaque élément de la hiérarchie. En prenant un indice de lien simple 5.3.

On peut associer, à toute hiérarchie indicée  $H$ , une mesure de dissimilarité qui s'appelle une *distance ultramétrique* [Cornuéjols et Miclet, 2010].

**Définition 5.5.3** Une *ultra métrique* (abrégé de *distance ultramétrique*) sur un ensemble  $H$  est une application de  $H \times H \rightarrow \mathbb{R}$  qui vérifie les trois conditions suivantes pour tous les  $h_i, h_j$  et  $h_k$  de  $H$  :

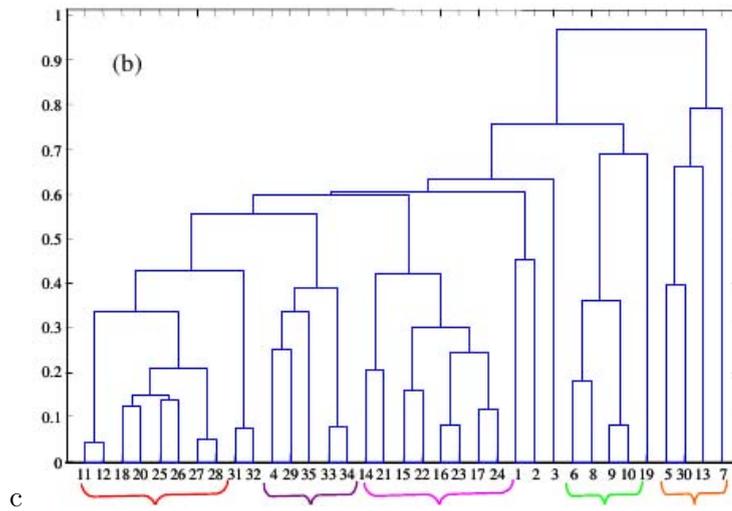


FIGURE 5.2 – Une hiérarchie construite pour la classification hiérarchique d’algorithmes de classification basée sur 35 indices [Jain *et al.*, 2004].

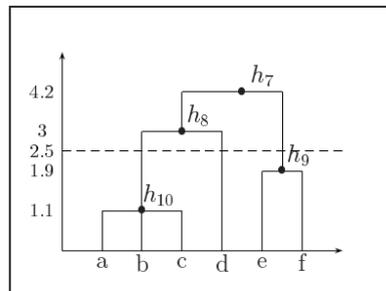


FIGURE 5.3 – Une hiérarchie indicée construite sur l’ensemble  $X$  en utilisant l’indice de lien simple.

- $\delta(h_i, h_i) = 0$  ;
- $\delta(h_i, h_j) = \delta(h_j, h_i)$  ;
- $\delta(h_i, h_k) = \max(\delta(h_i, h_j), \delta(h_j, h_k))$ .

L’Algorithme 6 illustre un algorithme général de classification hiérarchique qui prend en entrée un ensemble  $X$  sur lequel est définie une distance  $d$  entre ses éléments et une ultramétrie  $D$  pour la hiérarchie à construire. L’algorithme retourne une hiérarchie en fonction de la distance choisie. Nous allons décrire l’algorithme puis donner un certain nombre de distances ultramétriques couramment utilisées.

---

**Algorithme 6** : Algorithme de classification hiérarchique

---

Établir la table  $T_D$  des valeurs de  $D(x, y)$  pour  $x, y \in X$   
**tant que** La table  $T_D$  a plus d’une colonne **faire**  
 Choisir les deux sous-ensembles  $h_i, h_j$  de  $X$  tel que  $D(h_i, h_j)$  est le plus petit nombre réel dans la table  $T_D$   
 Supprimer  $h_j$  de la table, remplacer par  $h_i \cup h_j$   
 Calculer les mesures de similarité  $D$  entre  $h_i \cup h_j$  et les autres éléments de la table.  
**fin tq**

---

### 5.5.0.1 Les mesures de similarité

**La similarité du lien simple** mesure la distance entre les deux points les plus proches, l’un dans une classe et l’autre dans la seconde. La hiérarchie indiquée trouvée est monotone [Cornuéjols et Miclet, 2010].

Elle est donnée par la formule :

$$d_{min}(c_i, c_j) = \min_{p \in c_i, p' \in c_j} \{d(p, p')\} \quad (5.48)$$

De la même façon, une **similarité de lien multiple** détermine la distance entre les deux points les plus éloignés. Elle est donnée par :

$$d_{max}(c_i, c_j) = \max_{p \in c_i, p' \in c_j} \{d(p, p')\} \quad (5.49)$$

Les similarités du lien simple et multiple sont assez sensibles aux valeurs aberrantes et aux données bruitées. Cet inconvénient est contourné par l’usage de la distance entre les centres de gravité des classes et la moyenne des distances [Han *et al.*, 2006].

**La distance entre centres de gravité** détermine la distance euclidienne entre les centres de gravité des deux classes.

$$d_m(c_i, c_j) = d(m_i, m_j) \quad (5.50)$$

**La moyenne des distances** détermine la moyenne de toutes les distances entre les éléments des deux classes.

$$d_{md}(c_i, c_j) = \frac{1}{n_i n_j} \sum_{p \in c_i} \sum_{p' \in c_j} d(p, p') \quad (5.51)$$

L'indice de Ward permet de prendre en compte la variance des classes au moment de les fusionner.

Soient  $c_i$  et  $c_j$  les deux classes les plus proches. La similarité entre la nouvelle classe formée par la fusion de  $c_i$  et  $c_j$  avec une classe  $c_k$  est donnée par :

$$D(c_i \cup c_j, c_k) = \frac{n_k + n_i}{n_k + n_i + n_j} D(c_i, c_k) + \frac{n_k + n_j}{n_k + n_i + n_j} D(c_j, c_k) - \frac{n_i + n_j}{n_k + n_i + n_j} D(c_i, c_j) \quad (5.52)$$

Où  $n_i, n_j, n_k$  sont les nombres d'éléments dans les classes  $c_i, c_j, c_k$ . La dissimilarité  $D$  peut être une dissimilarité de lien simple.

### 5.5.1 Évaluation d'une hiérarchie

L'évaluation d'un algorithme hiérarchique peut se faire avec les indices donnés dans la section 5.4.3. Un autre moyen d'évaluer un algorithme de classification hiérarchique est le nombre de classes trouvées par l'algorithme. De façon générale, en classification hiérarchique, le nombre de classes dépend du moment où l'utilisateur décide d'arrêter le processus.

Les méthodes hiérarchiques souffrent du fait qu'on ne peut pas revenir sur la décision d'agglomérer ou de diviser. Néanmoins cette rigidité permet de limiter le temps de calcul en évitant les choix combinatoires. Deux approches sont utilisées pour y remédier :

1. choisir soigneusement le « liens » entre les objets au moment de la prise de décision.
2. procéder d'abord par classification hiérarchique pour former des classes initiales et appliquer une méthode de partitionnement.

Dans les deux cas, l'utilisateur peut spécifier le nombre de classes à trouver comme un critère d'arrêt.

Comme cela a été expliqué dans la section 5.4.5 des heuristiques sont utilisés pour estimer ce nombre.

## 5.6 Les méthodes probabilistes

Les méthodes probabilistes sont basées sur l'hypothèse que les classes ont été générées à partir d'une ou de plusieurs distributions de probabilité. Généralement, c'est un mélange de distributions gaussiennes qui est considéré. Le problème est d'estimer les paramètres de ces  $k$  distributions [Cornuéjols et Miclet, 2010].

L'algorithme Estimation-Maximisation permet d'estimer les paramètres de plusieurs modèles par exemple les Chaînes de Markov Cachées et les filtres bayésiens.

L'algorithme EM, donné par l'Algorithme 7, peut être appliqué à la classification automatique de données avec un fonctionnement similaire à celui de k-means, Dans le sens où les objets sont assignés aux classes en fonction d'une probabilité d'appartenance à cette classe, recalculée au fur et à mesure. Une différence importante est qu'il n'y a pas de frontière stricte entre les classes.

**Algorithme 7** : Estimation-Maximisation

Définir une statistique suffisante pour estimer l'ensemble  $\Pi$  des paramètres cachés

Initialiser  $\Pi$

$i \leftarrow 1$

**tant que**  $\Pi_i \neq \Pi_{i+1}$  **faire**

ESTIMATION

Utiliser les observations pour calculer la statistique suffisante de  $\Pi_p$

MAXIMISATION

Calculer  $\Pi_{i+1}$  comme une estimation au maximum de vraisemblance de  $\Pi$  à partir des résultats de l'étape  $p$  d'estimation

$p \leftarrow i + 1$

**fin tq**

Considérons que l'ensemble  $X = \{x_1, \dots, x_n\}$  soit généré par un mélange de  $k$  distributions gaussiennes  $N_1, \dots, N_k$ , dont on ne connaît pas les paramètres  $\mu_i, i = 1, \dots, k$ . Les variances ou les matrices de covariances peuvent être identiques ou différents. Ce sont ces paramètres que EM va estimer.

L'algorithme EM pour la classification est décrit par [Han *et al.*, 2006] :

1. Tirer aléatoirement  $k$  objets pour représenter le centre ou la moyenne des classes. D'autres paramètres de tendance centrale peuvent aussi être utilisés.
2. Raffiner de façon itérative les paramètres (ou les classes) en se basant sur ces deux étapes suivantes :
  - (a) ESTIMATION : Assigne chaque objet  $x_i$  à la classe  $c_k$  avec la probabilité

$$p(x_i \in c_k) = p(c_k | x_i) = \frac{p(c_k)p(x_i | c_k)}{p(x_i)} \quad (5.53)$$

où  $p(x_i) = N(\mu_k, E_k(x_i))$  suit la loi normale de moyenne  $\mu_k$ , et d'espérance  $E_k$ . En d'autres termes, cette étape calcule la probabilité d'appartenance d'un objet  $x_i$  pour chaque classe. Ces probabilités sont les valeurs « attendues » d'appartenance de l'objet  $x_i$ .

- (b) MAXIMISATION : Utiliser les estimations de probabilité à partir de l'étape précédente pour encore estimer (ou raffiner) les paramètres du modèle.

$$\mu_k = \frac{1}{n} \sum_{i=1}^n \frac{x_i P(x_i \in c_k)}{\sum_j P(x_i \in c_j)} \quad (5.54)$$

Cette étape est la maximisation de la vraisemblance de la distribution en considérant les données.

L'algorithme EM est simple et facile à implémenter. En pratique, il converge rapidement mais pourrait ne pas atteindre l'optimum global. La convergence est garantie pour certaines

formes de fonctions d'optimisation. La complexité est linéaire en fonction du nombre  $d$  de caractéristiques,  $n$  le nombre d'objets et  $t$  le nombre d'itérations [Han *et al.*, 2006].

AutoClass est un exemple d'algorithme de classification basé sur EM. Il combine EM avec un modèle bayésien et permet d'avoir de meilleurs résultats.

## 5.7 Les méthodes basées sur la densité

La plupart des méthodes de partitionnement utilisent une distance pour mesurer la dissimilarité entre les objets. De telles méthodes peuvent trouver des classes sphériques mais rencontrent des difficultés lorsqu'il s'agit de découvrir des classes de formes arbitraires.

Les méthodes présentées dans cette section sont basées sur l'hypothèse qu'une classe est une région dense dans l'espace de représentation et est séparée des autres classes par des régions moins denses. Nous allons décrire deux méthodes : DBSCAN (Density-Based Spatial Clustering of Applications with Noise) et DENCLUE (DENSITY-based Clustering). Le choix de ces méthodes a été guidé par le fait que DENCLUE a été hybridé par un algorithme de fourmis [Zaharie et Zamfirache, 2005] et que nous avons voulu exploiter certains mécanismes de DBSCAN pour proposer « une méthodes de chasse » pour les fourmis.

### 5.7.1 DBSCAN

L'idée de base de l'algorithme DBSCAN [Ester *et al.*, 1996] est que, pour tout point d'une classe, le voisinage d'un certain rayon doit contenir un nombre minimal de points. Pour construire les classes, la méthode est basée sur les concepts suivants :

- Le voisinage  $eps$  d'un point  $x_i$ ,  $N_{eps}(x_i)$ , est défini par

$$N_{eps}(x_i) = \{x_j \in X \mid d(x_i, x_j) \leq eps\}$$

Si le nombre d'objets dans  $N_{eps}(x_i) = MinPts$  alors  $x_i$  est un **core-object**. Le choix de la distance  $d$  conditionne [Ester *et al.*, 1996] la forme du voisinage et dépend de l'application. Deux types de points ont été définis : les points qui se trouvent à l'intérieur de la classe et ceux appartenant à la classe mais situés sur sa frontière. Les relations suivantes sont définies entre les points d'une même classe. Elles sont illustrées sur la Figure 5.4.

- un objet  $x_i$  est **directly density-reachable** à partir d'un objet  $x_j$ , en considérant  $eps$ ,  $MinPts$  si :
  - $x_i \in N_{eps}(x_j)$  ;
  - $x_j$  est un **core-object**.
- un objet  $x_i$  est **density reachable** à partir d'un objet  $x_j$  en considérant  $eps$  et  $MinPts$ , s'il existe une chaîne d'objets  $x_1, \dots, x_n$ , où  $x_1 = x_i$  et  $x_j = x_n$  tel que  $x_{i+1}$  est **directly density-reachable** à partir de  $p_i$  en considérant  $eps$  et  $MinPts$ , pour  $1 \leq i \leq n$ . Les points situés sur la frontière de la classe peuvent ne pas être **density-reachable**. Une relation liant ces points en passant par un **core-object** est donnée par :

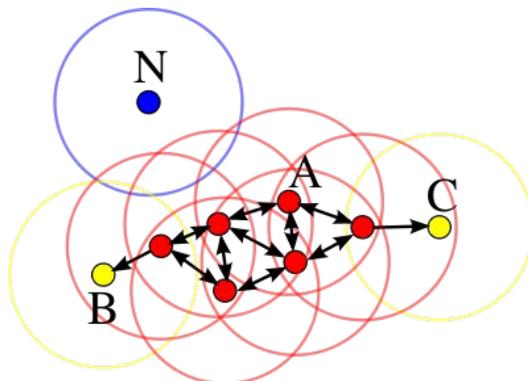


FIGURE 5.4 – Le point A est un core-object à partir duquel les points B et C sont density-reachable. Donc A, B, C appartiennent à la même classe. Le point N est un bruit.

- un objet  $p$  est *density-connected* à un objet  $q$ , s'il existe un point  $o$  tel que  $p$  et  $q$  sont tous les deux *density-reachable* à partir de  $o$  en considérant  $eps, MinPts$ . La relation *density-connectivity* est symétrique et réflexive pour des points *density-reachable*.
- une classe  $C$ , considérant  $eps$  et  $MinPts$  est un sous-ensemble non vide de  $X$  satisfaisant aux conditions suivantes :
  - i)  $\forall p, q$  : si  $p \in C$  et  $q$  est *density-reachable* à partir de  $p$  alors  $q \in C$  ; Maximalité
  - ii)  $\forall p, q \in C$  :  $p$  est *density-connected* à  $q$ . Connectivité.
 Une classe  $C$  contient au minimum  $MinPts$  objets pour la raison que le premier objet de la classe est un *core-object*.
- Un bruit est tout point n'appartenant pas à une classe.

Pour trouver les classes, DBSCAN commence par un point arbitraire  $p$  et détermine tous les points qui sont *density-reachable* à partir de  $p$ . Si  $p$  n'a pas été déjà classé et est un core point alors une classe est créée. Ainsi, tous les points qui sont dans le voisinage de  $p$  sont successivement évaluée pour étendre la nouvelle classe avec les objets qui sont dans leur voisinage.

Dans le cas où le  $eps$ -voisinage de  $p$  n'a pas  $MinPts$  objet,  $p$  est considéré être sur la frontière d'une classe et la méthode passe au prochain point dans la base de données.

Lorsque deux classes de différentes densités sont assez proches, elles peuvent être fusionnées en Considérant une distance de lien simple  $D$  entre deux ensembles de points d'une même classe, si cette distance est supérieure à  $eps$  alors la classe peut être découpée en deux sous-classes. Ainsi, DBSCAN est récursivement appliqué sur les classes ayant une valeur grande pour  $MinPts$ .

La détermination des valeurs des paramètres n'est pas évidente. Lorsque les valeurs de  $eps, MinPts$  sont connues pour chaque classe ainsi que le point  $p$  de départ alors les classes sont déterminées facilement par la méthode. Mais en général, ces paramètres ne sont pas connus et  $eps$  et  $MinPts$  sont déterminés à partir de la classe la moins dense de la base de données. Ensuite l'algorithme est initialisé par ces deux valeurs qui sont utilisées pour toutes les classes. Une heuristique a été utilisée pour déterminer  $eps, MinPts$ , mais, selon les auteurs, le choix reste toujours difficile.

La méthode souffre néanmoins de ses deux paramètres dont le choix est déterminant pour les classes trouvées.

### 5.7.2 DENCLUE

DENCLUE [Hinneburg et Gabriel, 2007] est basée sur un ensemble de distribution de fonctions de densité. La méthode est construite autour des idées suivantes :

- l'influence de chaque objet peut être modélisée formellement en utilisant une fonction mathématique appelée *influence function*. Cette fonction décrit l'impact d'un objet sur son voisinage ;
- la fonction de densité appropriée à l'ensemble des données peut être modélisée analytiquement comme étant la somme des fonctions d'influence appliquées à toutes les données ;
- les classes peuvent être déterminées en identifiant pour chaque classe un *density attractor* qui est un maximum local de la fonction de densité.

Soient  $x_i$ , et  $x_j$  deux objets de  $X$ . La fonction d'influence de  $x_j$  sur  $x_i$  est une fonction définie de  $f_B^{x_j} : X \rightarrow R^+$  qui est définie par la fonction basique  $f_B$  qui reflète l'impact de  $x_j$  sur  $x_i$  :

$$f_B^{x_j}(x) = f_B(x_j, x_i) \quad (5.55)$$

Une fonction influence peut être n'importe quelle fonction pouvant être déterminée par la distance entre deux objets dans un voisinage.

La fonction *square wave influence* est un exemple de fonction d'influence :

$$f_{square} = \begin{cases} 0 & \text{si } d(x_i, x_j) > \sigma \\ 1 & \text{sinon} \end{cases} \quad (5.56)$$

Si  $x_i$  et  $x_j$  sont « éloignés » l'un de l'autre alors la distance  $d(x_i, y_j)$  sera supérieure à  $\sigma$ . Dans ce cas la fonction d'influence vaut 0, et symbolise le manque d'influence entre les deux points. D'un autre côté si  $x_i$  et  $x_j$  sont « proches » (la proximité est déterminée par le paramètre  $\sigma$ ), une valeur de 1 est retournée, représentant l'influence de  $x_i$  sur  $x_j$ .

Une autre fonction est le noyau gaussien :

$$f_{Gauss}(x_i, x_j) = e^{-\frac{d(x_i, x_j)^2}{2\sigma^2}} \quad (5.57)$$

La fonction de densité d'un objet  $x_i \in X$  est définie par la somme des fonctions d'influence de tous les autres objets sur  $x_i$ . Soit l'ensemble des objets, la fonction de densité est :

$$f_B^d = \sum_{i=1}^n f_B^{x_i} = f_B^{x_1} + \dots + f_B^{x_n} \quad (5.58)$$

A partir de la fonction de densité, nous pouvons définir le gradient de la fonction et un attracteur de densité (le maximum local) pour la fonction de densité. Un point  $x_i$  est dit attiré par la densité par un attracteur de densité  $x^*$ , s'il existe un ensemble de points  $x_0, \dots, x_k$  tel que  $x_0 = x_i, x_k = x^*$  et le gradient de  $x_{i-1}$  est la direction de  $x_i$  pour  $1 \leq i \leq k$ . Intuitivement, un attracteur de densité influence plusieurs points. Pour les

fonctions continues et dérivables, un algorithme *hill climbing* guidé par le gradient peut être utilisé pour déterminer l'attracteur de densité d'un ensemble de points.

En général, les points qui sont attirés par la densité par  $x^*$  pourraient former une classe. Basées sur les notions précédentes deux types de classes ont été définis : les classes définies par des centres, les classes de forme arbitraire.

Une classe définie par un centre, pour un attracteur de densité  $x^*$ , est un sous-ensemble de points  $C \subseteq X$ , qui sont attirés par la densité par  $x^*$  et où la fonction de densité au niveau de  $x^*$  est au moins supérieure à un seuil  $\xi$ . Les points dont la densité est supérieure à  $\xi$  sont considérés comme des valeurs aberrantes. Les points dans une classe sont influencés par plusieurs autres points, alors que les valeurs aberrantes ne le sont pas.

Une classe de forme arbitraire pour un ensemble d'attracteurs de densité est un ensemble de points dont chacun est attiré par la densité par son attracteur de densité où : 1) la valeur de la fonction de densité est supérieure ou égale à  $\xi$  ; 2) il existe un chemin,  $P$  à partir de chaque attracteur de densité à un autre, où la valeur de la fonction de densité pour chaque point sur le chemin est supérieure ou égale à  $\xi$ .

Les principales avantages de Denclue sont [Han *et al.*, 2006] : 1) ses bases mathématiques et la généralisation de plusieurs algorithmes de classification (k-means et DBSCAN) ; 2) il est adapté pour la classification de données ayant beaucoup de bruit ; 3) il permet une description compacte des classes de forme arbitraire dans de grandes dimensions.

Mais la méthode requiert un choix minutieux des paramètres  $\sigma$  et du seuil  $\xi$ . Ces choix peuvent influencer les performances de façon significative.

## 5.8 La classification par fourmis artificielles

### 5.8.1 Les origines

L'intelligence collective (Swarm Intelligence) est un paradigme basé sur le comportement de certains organismes vivant en colonie. Ces comportements ont montré leur efficacité dans la nature de part leur simplicité, auto-gestion, auto-organisation, leur évolutivité et robustesse. Naturellement, ces méthodes ont été, et continuent d'être, la source d'inspiration de plusieurs heuristiques pour résoudre des problèmes informatiques difficiles. Néanmoins, dans cette partie, nous allons nous focaliser sur la classification par fourmis artificielles.

La classification de données par fourmis artificielles trouve son origine dans un modèle développé par Jean louis Deneubourg *et al.* [Deneubourg *et al.*, 1991] pour simuler le comportement d'une colonie de fourmis par un groupe de robots (assez simples). L'idée est de montrer que des agents (robots) qui ne communiquent pas directement entre eux, ne disposent pas d'une représentation globale de l'environnement (une carte de l'emplacement des objets), et fonctionnent de façon décentralisée, sont capables d'imiter le comportement de tri d'une colonie de fourmis. Les robots ont cependant la capacité de distinguer entre deux ou plusieurs types d'objets et surtout de prendre et de déposer un objet. Cette approche, bien que moins efficace qu'un comportement hiérarchique offre les avantages de la simplicité, la flexibilité et la robustesse [Deneubourg *et al.*, 1991].

Deux types de tri ont été observés dans les colonies de fourmis : le tri des tâches

(*patch sorting*) et le tri annulaire. Comme les noms le suggèrent, le tri des tâches arrange simplement les objets en tâche (voisinage spatial) alors que le tri annulaire arrange les objets en des anneaux (concentriques) complets ou partiels.

Le tri de tâches est utilisé par les fourmis *Pheidole pallidula* dans la formation des cimetières. Il a été observé que les fourmis de cette colonie classifient les cadavres des autres fourmis pour les sortir du nid. Ce tri est la base du modèle DM qui sera détaillé dans la section 5.8.2.

Le tri annulaire [Deneubourg *et al.*, 1991, Sendova-Franks et Franks, 1999] a été observé chez les fourmis *Leptothorax unifasciatus* qui arrangent leurs larves en fonction de leurs tailles.

Le modèle de DM a donné lieu à plusieurs algorithmes de classification. Ces méthodes peuvent être regroupées en trois [Handl et Meyer, 2007] grandes catégories :

1. La première catégorie groupe les méthodes dérivées du modèle DM et imitent le comportement naturel des fourmis. Une caractéristique de ces méthodes est que l'objectif est implicitement défini. Implicite ici veut dire aussi bien un objectif global pour le clustering que les types de classes à trouver ne sont définis explicitement à aucun moment durant le processus de classification. Une évaluation de l'adaptabilité de telles méthodes pour la classification d'un type de données particulier peut être un défi ;
2. Pour le second groupe, le problème de classification est redéfini en problème d'optimisation et résolu par un algorithme d'optimisation. Il faut remarquer que le comportement des fourmis a aussi été exploité pour l'optimisation combinatoire. Dans ce cas, c'est, entre autres, la modification du milieu de chasse par dépôt d'une substance chimique (phéromone) ou stigmergie qui est exploité. Ant Colony Optimization (ACO) [Bonabeau *et al.*, 1999] est une méthode qui a donné de meilleurs résultats et est très populaire. Un avantage de cette approche est la définition explicite de la fonction objectif, qui permet de mieux comprendre et de prédire les performances de l'algorithme sur un type particulier de données et de paramétrer la fonction objectif pour une tâche particulière ou un type de données.
3. Un certain nombre d'algorithmes par contre, se situent entre ces deux groupes. : Ils sont dérivés de méthodes d'optimisation, comme ACO, mais modifiées à un point qu'aucune fonction objectif n'existe. D'autres algorithmes sont des hybridations avec les autres paradigmes présentés plus haut (partitionnement, hiérarchique, probabiliste, basé sur la densité).

Dans nos travaux, nous ne nous sommes pas intéressés au fait de savoir laquelle de ces catégories produit les meilleurs résultats. Au vu des divers travaux dans ces domaines, ni la proximité de la méthode avec la nature ni l'optimisation n'influence en rien les résultats. Cependant, il est important de mieux choisir les paramètres de l'heuristique afin de l'adapter aux données et à l'application. Même si par rapport aux autres paradigmes, l'application réelle des méthodes de classification par fourmis artificielles est moindre.

### 5.8.2 Le modèle de Deneubourg

Le modèle DM est basé sur deux règles. Soient des agents (des robots) qui se déplacent aléatoirement dans la zone de simulation [Deneubourg *et al.*, 1991] :

- lorsqu'ils rencontrent un objet, la probabilité  $p$ , qu'ils le prennent augmente d'autant plus que cet objet est isolé ;
- lorsqu'ils portent un objet, la probabilité  $d$ , de le déposer est d'autant plus grande que des objets similaires sont dans le voisinage immédiat de l'agent.

C'est la combinaison de ces deux comportements qui va provoquer au fil de la simulation un tri des objets en un tas (lorsqu'il n'y a qu'un seul type d'objets) et en autant de tas lorsqu'il en a plusieurs types. Quand un tas d'objets similaires apparaît par « chance », il va attirer les agents porteurs d'objets qui passent à côté pour déposer leur charge et cela augmente l'attractivité de ce tas.

Les deux probabilités sont données par :

$$p = (k_p / (k_p + f))^2 \quad (5.59)$$

$$d = (f / (k_d + f))^2 \quad (5.60)$$

Où  $k_p$  et  $k_d$  sont des constantes et  $f$  est une estimation de la fraction des objets de même type qui sont dans le voisinage. L'usage de  $f$  va de paire avec celle d'une mémoire,  $m$ , des étapes précédentes dont dispose chaque fourmi.

Par exemple [Deneubourg *et al.*, 1991], si dans la zone de simulation, il y a deux types d'objets  $A$  et  $B$  et que la taille de la mémoire est 10. A un moment donné, la mémoire pourrait contenir 00AB0AA0B0. Cela indique que durant les 10 pas précédents, le robot a rencontré 3 objets de type  $A$  et 2 de type  $B$ , le 0 indique qu'aucun objet n'a été rencontré. Ainsi  $f_A = 3/10$  et  $f_B = 2/10$ . Pour la probabilité  $p$ , plus  $f$  est grande (tend vers 1) moins  $p$  augmente et inversement pour la probabilité  $d$ .

#### 5.8.2.1 Le choix des paramètres

Une remarque importante concernant le paramétrage du modèle est donnée par les auteurs : "*Les valeurs des paramètres utilisées ont été sélectionnées de façon plus ou moins aléatoire aidé en cela par le fait que le modèle trie efficacement avec un intervalle assez large de valeurs pour ces paramètres*" [Deneubourg *et al.*, 1991]. Les valeurs qui ont été choisies sont :

- $k_p = 0.1$  ;
- $k_d = 0.3$  ;
- $m = 15, 50$ .

Une remarque à propos de  $m$  est qu'une grande valeur empêche un tri efficace car elle empêche les fourmis de distinguer un objet isolé de celui qui ne l'est pas et qui est déjà dans un tas [Deneubourg *et al.*, 1991].

Les valeurs données pour les paramètres ont été reprises par la suite dans d'autres études sans qu'il ait été montré l'impact de ces paramètres sur le modèle.

### 5.8.2.2 Les bases de la classification et les espoirs pour le topographic mapping

En réalité, bien que le mot classification a été utilisé à plusieurs reprises, les agents ne font que trier des objets au maximum de deux types. Il n'en demeure pas moins que les auteurs ont jeté les bases qui seront reprises par Lumer et Faieta [Lumer et Faieta, 1994] pour être généralisées à une classification automatique de données.

Cette idée est la suivante. Elle consiste à introduire un chevauchement ou une imperfection entre les deux types d'objets [Deneubourg *et al.*, 1991]. Lorsqu'un agent non chargé rencontre un objet de type  $A$ , il va déterminer  $f_A$  en introduisant une erreur  $e$  multipliée par le nombre d'objets de type  $B$ . On a alors :

$$f_A = \frac{n_A + n_B * e}{m} \quad (5.61)$$

ou  $n_A$  et  $n_B$  sont le nombre d'objets de type A et B. Cette erreur permet aux agents de pouvoir former des tas dispersés ou proches assez rapidement ou lentement. Il peut aussi permettre une spécialisation de certains agents pour un type d'objet et apporter une diversification dans la recherche des objets par les agents dans leur voisinage. Selon les auteurs, les agents sont capables de placer des tas d'objets de types différents l'un à côté de l'autre (en fonction de leur différence) avec l'usage du facteur d'erreur. Nous pensons que c'est cela qui donnera l'espoir à Lumer et Faieta [Lumer et Faieta, 1994] que les agents sont capables de classer les objets et en faisant du *topographic mapping*.

### 5.8.3 Le modèle de Lumer et Faieta

Le modèle de Lumer et Faieta [Lumer et Faieta, 1994], LF, est la première adaptation du modèle DM à la classification de données. Ce modèle a eu le même objectif que celui de Deneubourg en ce sens que les auteurs ont plus illustré la possibilité d'appliquer des algorithmes de fourmis sur un problème de classification que de faire une classification de données. L'idée principale a été la définition d'une fonction de densité permettant de prendre en compte des objets à plusieurs dimensions et ainsi de pouvoir s'attaquer à plusieurs problèmes de classification. Cependant, au niveau de l'évaluation aucune mesure pertinente n'est utilisée pour savoir si le partitionnement obtenu est bon ou mauvais.

#### 5.8.3.1 Les conditions du modèle

Le modèle utilise des fourmis qui se déplacent sur une grille à deux dimensions sur laquelle les éléments sont dispersés aléatoirement. Une cellule de la grille ne contient qu'un seul élément. La taille de la grille est choisie de sorte que le nombre de cellules dépasse le nombre d'objets d'un certain ordre de grandeur.

Le nombre d'objets dépasse le nombre de fourmis d'un autre ordre de grandeur afin que les actions entreprises par les fourmis transforment petit à petit l'environnement (comportement émergent).

La simulation se fait de façon discrète : à chaque itération une fourmi est choisie aléatoirement et peut prendre un objet (si la fourmi en trouve à sa position courante) ou déposer un objet (si elle est chargée). Donc, en une seule itération, la prise n'est pas suivie d'un

dépôt. Mais la fourmi prend puis à la prochaine itération où elle est choisie, elle dépose l'objet. Ce qui, entre autre, ralentit la convergence. À la fin de l'itération la fourmi se déplace dans une des cellules voisines avec la même probabilité.

En considérant qu'une fourmi ne porte pas d'objet, la probabilité qu'elle prenne un objet augmente avec une faible densité de cet objet dans son voisinage. Cette probabilité de ramasser l'objet  $i$  est donnée par :

$$P_{pickup} = \left( \frac{k_p}{k_p + f(i)} \right)^2 \quad (5.62)$$

$v$  est défini comme un carré de  $d \times d$ .  $k_p$  est une constante et  $f(i)$  est une estimation locale de la densité des objets et de leur similarité.

De la même manière, la probabilité de dépôt devrait augmenter lorsque, dans le voisinage de l'objet  $i$ , la densité de cet objet est grande.

$$P_{drop} = \begin{cases} 2 \times f(i) & \text{si } f(i) < k_d \\ 1 & \text{sinon} \end{cases} \quad (5.63)$$

$k_d$  est une autre constante.

La fonction  $f(i)$  qui mesure la similarité et la densité est donnée par :

$$f(i) = \max \left\{ 0, \frac{1}{d^2} \sum (1 - d(i, j)/\alpha) \right\} \quad (5.64)$$

$\alpha$  est un paramètre de normalisation des données.

### 5.8.3.2 Les données utilisées

Les tests montrés dans [Lumer et Faieta, 1994] ont été faits avec une base de 400 données artificielles à 2 dimensions. Ces données ont été générées en utilisant quatre distributions gaussiennes définie par :

$$(N(0, 2), N(0, 2)), (N(0, 2), N(8, 2)), (N(8, 2), N(0, 2)), (N(8, 2), N(8, 2))$$

. Cela représente des classes assez bien séparées.

### 5.8.3.3 Les paramètres du modèle

Le paramètre  $\alpha$  est très important pour ce modèle. Il constitue une échelle pour la dissimilarité dont le choix détermine les classes trouvées par l'algorithme.  $d^2$  est un terme de normalisation dans le voisinage considéré et introduit une dépendance à la densité dans la fonction  $f(i)$ . Ainsi, le maximum de  $f(i)$  est atteint si tous les éléments dans le voisinage sont identiques ( $d(i, j) = 0$ ). Le nombre de fourmis a été fixé à 40 qui est 1/10<sup>ème</sup> du nombre de données. Les paramètres  $k_p$  et  $k_d$  ont les même valeurs que pour DM.

### 5.8.3.4 L'algorithme

En résumé l'algorithme boucle sur trois étapes :

1. Sélection d'une fourmi de façon aléatoire ;
2. Si faisable, manipuler l'objet (prendre ou déposer) ;
3. Un déplacement aléatoire sur la grille.

L'algorithme est décrit en langage évolué sur l'Algorithme 8.

---

**Algorithme 8** : L'algorithme de Lumer et Faieta

---

```
Placer aléatoirement les objets et les fourmis sur la grille /*La boucle principale */
pour  $T = 1$  à  $T_{max}$  faire
    Tirer aléatoirement une fourmi  $a_i$  dans l'ensemble des fourmis si La fourmi  $a_i$ 
ne transporte pas d'objet alors
    | si La fourmi  $a_i$  trouve un objet  $i$  à sa position courante alors
    | | calculer  $f(i)$  et  $P_{pickup}(i)$  La fourmi tente de ramasser l'objet suivant la
    | | probabilité  $P_{pickup}(i)$ 
    | fin si
    fin si
    sinon
    | /* La fourmi transporte un objet  $i$  */ Calculer  $f(o_i)$  et  $P_{drop}(i)$  Tenter de
    | déposer l'objet sur la cellule courante avec la probabilité  $P_{drop}(i)$  Ou se
    | déplacer dans une cellule voisine pour déposer l'objet
    fin si
    Déplacer la fourmi  $a_i$  dans son voisinage sur une cellule non occupée par une
    autre fourmi
fin pour
```

---

### 5.8.3.5 Observations

Un des inconvénients de la méthode LF est qu'elle ne retourne aucun partitionnement des données à l'image par exemple de k-means. De ce fait, les auteurs n'ont implémenté aucun mécanisme permettant de vérifier la qualité d'un partitionnement et de pouvoir comparer les résultats avec d'autres méthodes. La visualisation des tas d'objets sur la grille, montre que le nombre de classes formées par les fourmis sur la grille dépasse celui existant dans les données.

Une autre remarque importante a été faite par les auteurs à propos du rôle du paramètre  $\alpha$ . Elle est la suivante : "*La taille de l'échantillon de données utilisée devrait être suffisamment large pour produire des ensembles localement denses de données similaires. Ce qui veut dire que le nombre moyen de points différents d'un autre point de  $\alpha$  devrait être de l'ordre de la surface  $d^2$  d'un agent se déplaçant sur la grille.*" [Lumer et Faieta, 1994].

Nous formulons cette remarque par :

$$\forall x \in S, \|x - x_i\| \leq \alpha$$

$S$  est un ensemble de points dont la taille est de l'ordre de  $d^2$ .  $x_i$  est un objet de  $S$ .  $\alpha$  permet à une fourmi de pouvoir détecter les objets qui sont similaires à un certain degré dans son voisinage : donc les objets qui ne respectent pas cette considération ne sont pas pris en compte. De ce fait, les auteurs ont introduit un certain nombre de modifications qui seront largement reprises par la suite par d'autres auteurs.

### 5.8.3.6 Modifications

Trois modifications ont été proposées [Lumer et Faieta, 1994] :

1. diversité de la population : utilisation d'une population de fourmis avec diverses vitesses de déplacement et une sensibilité à la similarité des objets. Cette proposition sera reprise dans plusieurs travaux sous la forme d'un  $\alpha$  adaptatif ;
2. mémoire : mémoriser les plus récents succès de dépôt pour permettre à la fourmi de déposer les objets. Plusieurs travaux ont aussi repris cette proposition en mettant en place des politiques de remplacement de la mémoire, par exemple : FIFO, etc ;
3. Changement de comportement : les individus changent de comportement pour impacter sur le phénotype. Ceci sera détailler plus en avant.

**Diversité de la population** : Cela consiste à rajouter une caractéristique de vitesse pour les fourmis allant de 1 à  $V_{max}$ . Ainsi la population contient deux types de fourmis :

- rapide : la valeur du rayon de perception est large. elles vont loin sur la grille et prennent plus souvent des objets dont la similarité peut être faible ;
- lente : elles restent dans le voisinage pour trier plus finement les objets ramenés par les fourmis rapides ;

Suivant cette modification, la fonction  $f(i)$  devient :

$$f(i) = \max \left\{ 0, \frac{1}{d^2} \sum \left( 1 - \frac{d(i, j)}{\alpha + \alpha(v - 1)/vmax} \right) \right\} \quad (5.65)$$

Pour chaque fourmi le choix de la vitesse est fait uniformément entre [1, 6].

**Mémoire à court terme** : Une mémoire est utilisée, contenant les  $m$  plus récents objets déposés par la fourmi avec leurs positions. La fourmi ayant pris un objet la compare avec ceux de sa mémoire et dépose l'objet près de celui qui est le plus similaire. La taille de la mémoire joue sur les performances, comme cela a été décrit pour DM.

**Changement de Comportement** : il consiste, pour les fourmis de détruire certaines classes, lorsque les déplacements deviennent moins fréquents. En effet, plus la simulation avance, moins les objets sont déplacés par les fourmis.

Nous venons de donner les bases utilisées par la majorité des algorithmes de fourmis utilisés pour la classification automatique de données.

## 5.9 Conclusion

Dans ce chapitre nous avons présenté la classification automatique de données en expliquant, les difficultés liés à cette tâche et aussi les divers paradigmes existants. Nous avons

## 5.9. CONCLUSION

---

aussi présenté une partie importante du processus de classification qui est l'évaluation et la comparaison d'un ou de plusieurs partitionnements. La classification par fourmis artificielles est relativement récente comparée aux méthodes de partitionnement par exemple. Elle a donné lieu à plusieurs travaux. Dans le chapitre suivant nous allons présenter notre contribution et nous baser sur les critères d'évaluation données ici pour l'évaluer.

## Chapitre 6

# API : Les méthodes de chasses des fourmis *Pachycondyla apicalis* appliquées au clustering

### 6.1 Introduction

L'algorithme API est basé sur le comportement de recherche de nourriture des fourmis *Pachycondyla apicalis*. Il a été initialement proposé pour l'optimisation continue par Nicolas Monmarché dans sa thèse [Monmarché, 2000]. API est considéré comme une méta-heuristique et peut être appliqué à des espaces de recherche variés, comme les espaces discrets ou mixtes. Un des avantages de API est qu'il est basé sur une espèce particulière de fourmis. Le comportement des fourmis *Pachycondyla apicalis* est intéressant parce qu'il est simple et assez efficace par rapport à l'environnement de ces fourmis. Nous suggérons que cette simplicité peut cependant conduire à des idées fertiles, non seulement pour l'optimisation, mais aussi dans le domaine de la classification de données. Plusieurs essais ont été effectués pour transposer les divers comportements de fourmis pour attaquer le problème de classification de données, mais, à notre connaissance, l'algorithme API pour le clustering n'a été étudié qu'une seule fois et d'une façon différente de celle que nous allons présenter [Admane *et al.*, 2006]. Les travaux que nous allons présenter ont été publiés dans [Amadou Kountché *et al.*, 2012].

### 6.2 API pour la classification de données

#### 6.2.1 Principes de API

API est, comme nous l'avons déjà mentionné, basé sur le comportement des fourmis *Pachycondyla apicalis*. Ces fourmis ont la capacité de chercher de la nourriture et d'exploiter des sources distribuées sans aucun comportement centralisé et sans recrutement de masse, assez souvent rencontré chez plusieurs espèces de fourmis. Au lieu de cela, les fourmis *P. apicalis* ont un comportement simple pour apporter assez de proies pour la survie

de la colonie. Nous pouvons décrire ce comportement avec quelques règles qui pourront facilement être traduites en algorithme :

1. *P. apicalis* cherchent de la nourriture autour de leur nid (notion de point central et exploration aléatoire) ;
2. une fourmi *P. apicalis* mémorise la place où elle a attrapé sa proie (généralement un autre insecte, mort ou vif) et retourne directement à son nid (pas de dépôt de phéromone) ;
3. à sa prochaine sortie, une fourmi *P. apicalis* retourne à son dernier lieu de chasse où elle a réussi à attraper une proie et cherche une nouvelle proie à partir de cette position (notion d'exploration locale) ;
4. les fourmis *P. apicalis* ne construisent pas leur propre nid et sont, par conséquent, obligées de déplacer leur colonie lorsque les conditions de vie ne sont pas pérennes (opérateur de redémarrage) ;
5. lorsque la colonie décide de se déplacer, les ouvrières utilisent une forme simple de recrutement, appelé *tandem running* : une fourmi conduit les autres à la nouvelle position du nid et ainsi de suite jusqu'à ce que toute la colonie soit au courant de la nouvelle position du nid.

Le modèle que nous pouvons déduire à partir de ces règles contient des agents dans un espace de recherche dans lequel les points correspondent à des solutions du problème étudié. En l'occurrence, pour un problème d'optimisation continue, l'espace de recherche peut être un sous-espace  $S$  de  $\mathcal{R}^n$  dans lequel nous recherchons une valeur minimale atteinte par une fonction inconnue  $f$ , définie sur ce sous-espace. L'algorithme API, dans ce cas, consiste à démarrer à une position aléatoire de  $S$  pour le nid. Puis, les fourmis créent des sites de chasse aléatoirement autour du nid. À chaque itération, chaque fourmi retourne à son site de chasse et opère une recherche locale aléatoire. Si cette tentative est un succès, on considère que la fourmi a trouvé une proie. Sinon, plusieurs visites sans succès au même site de chasse peuvent décourager la fourmi qui va recommencer avec un nouveau point généré autour du nid.

La version de API, décrite ci-dessus, est un algorithme stochastique d'optimisation basé sur une population : les sites de chasse mémorisés par les fourmis correspondent à la population et le nid joue un rôle d'échange d'information entre les éléments de la population de solutions. Dans ce qui suit, nous allons nous focaliser sur API pour la classification de données.

### 6.2.2 Notations et conditions initiales

Nous avons défini quatre ensembles :

- $\mathcal{O} = \{\mathcal{O}_1, \dots, \mathcal{O}_O\}$  est l'ensemble des  $O$  objets (*i.e.* la base de données d'entrée) ;
- $\mathcal{A} = \{\mathcal{A}_1, \dots, \mathcal{A}_A\}$  est l'ensemble des  $A$  fourmis ;
- $\mathcal{N} = \{\mathcal{N}_1, \dots, \mathcal{N}_N\}$  est l'ensemble des  $N$  nids ;
- $\mathcal{G}$  est une grille à deux dimensions (qui peut être toroïdale ou non, assez souvent carrée) composée de cellules, chacune ayant 4 voisins dans le cas toroïdal.

	$\mathcal{O}_7$					
		•	•	$\mathcal{O}_5$	$\mathcal{O}_{10}$	
	•	$\mathcal{O}_2$	$\mathcal{O}_3$	•		
		•	$\mathcal{O}_4$	•		$\mathcal{O}_1$
		•	$\mathcal{O}_6$	•		
$\mathcal{O}_8$			•	$\mathcal{O}_9$		

FIGURE 6.1 – Dans une grille  $7 \times 7$ , l'ensemble des cellules voisines vides d'un objet  $\mathcal{O}_3$ , nommé  $\mathcal{E}(\mathcal{O}_3)$ , est représenté par un ensemble de • sur la figure. L'ensemble  $v(\mathcal{O}_3) = \{\mathcal{O}_2, \mathcal{O}_4\}$  dénote l'ensemble des objets qui sont des voisins directs de  $\mathcal{O}_3$ . L'ensemble  $\mathcal{V}(\mathcal{O}_3)$  correspond au quatre cellules environnantes.

Tous les ensembles  $\mathcal{O}$ ,  $\mathcal{A}$ ,  $\mathcal{N}$  sont positionnés sur la grille  $\mathcal{G}$ .

Nous avons défini les notations suivantes :

- $|\mathcal{N}_i|$  dénote le nombre d'objets qui appartiennent au nid  $\mathcal{N}_i$  ;
- $d(\mathcal{O}_i, \mathcal{O}_j)$  est la distance entre les deux objets. La distance est une distance euclidienne dans l'espace des paramètres.
- $\mathcal{V}(X)$  correspond au voisinage de Von Neumann de  $X$ , où  $X \in \mathcal{O} \cup \mathcal{A} \cup \mathcal{N} \cup \mathcal{G}$  ;
- $\mathcal{E}(X)$  correspond au voisinage étendu des cellules vides de  $X$  ( $X \in \mathcal{O} \cup \mathcal{A} \cup \mathcal{N} \cup \mathcal{G}$ ).  $\mathcal{E}(X) = \{cells = \phi \in \mathcal{V}(Y), Y \in \mathcal{V}(X)\}$ . Voir figure 6.1 ;
- $v(X)$  dénote l'ensemble des objets qui sont dans le voisinage direct (i.e dans le voisinage de Von Neumann) de  $X$  ( $X \in \mathcal{O} \cup \mathcal{A} \cup \mathcal{N} \cup \mathcal{G}$ ). Voir figure 6.1.

Premièrement, la grille toroïdale est construite avec au moins  $4 \times O$  cellules et les nids sont régulièrement dispersés sur la grille de façon à ce qu'il y ait plusieurs cellules vides entre chaque couple de nids (pour que les nids soient séparés dès le départ). Ensuite, les objets  $\mathcal{O}$  sont aléatoirement dispersés sur la grille (avec au maximum un objet par cellule). Les objets ne peuvent pas être initialement positionnés sur un nid.

A fourmis sont équitablement assignées au  $N$  nids (ainsi,  $A/N$  fourmis sont disponibles pour chaque nid) et sont assignées à des positions aléatoires sur la grille.

### 6.2.3 Description du comportement des agents

Une fois les objets, nids et fourmis, initialisés, l'algorithme simule le comportement des fourmis : à chaque étape, les fourmis peuvent se déplacer et faire des actions (prendre et déposer un objet). Une fourmi peut se déplacer de sa cellule courante à une de ces quatre cellules voisines et ses décisions sont gouvernées par des probabilités que nous allons décrire.

**Le comportement d'une fourmi pour prendre un objet.** Si une fourmi trouve un objet (i.e l'objet est sur la même cellule que la fourmi), alors le comportement dépend de l'objet :

- si l'objet est libre (i.e il n'appartient à aucun nid) ou si sa similarité avec son nid est

pire qu'avec le nid de la fourmi, alors la fourmi prend l'objet  $\mathcal{O}_i$  selon la probabilité :

$$P_{\text{pickup}}(\mathcal{O}_i, \mathcal{N}_j) = \left( \frac{f^{\text{pickup}}(\mathcal{O}_i, \mathcal{N}_j)}{k_p + f^{\text{pickup}}(\mathcal{O}_i, \mathcal{N}_j)} \right)^2 \quad (6.1)$$

Où  $\mathcal{N}_j$  est le nid auquel appartient la fourmi,  $k_p$  est un paramètre de l'algorithme, et  $f^{\text{pickup}}$  est une fonction de densité calculée en fonction des distances entre l'objet  $\mathcal{O}_i$  et les objets déjà dans le nid  $\mathcal{N}_j$  :

$$f^{\text{pickup}}(\mathcal{O}_i, \mathcal{N}_j) = \max \left\{ \frac{1}{|\mathcal{N}_j|} \sum_{\mathcal{O}_k \in \mathcal{N}_j} 1 - \frac{d(\mathcal{O}_i, \mathcal{O}_k)}{\alpha}; 0 \right\} \quad (6.2)$$

Où  $|\mathcal{N}_j|$  est le nombre d'objets appartenant à  $\mathcal{N}_j$ ,  $\alpha$  est une constante et un paramètre de l'algorithme. Il faut noter que si le nid  $\mathcal{N}_j$  est vide (i.e.  $|\mathcal{N}_j| = 0$ ) la probabilité  $P_{\text{pickup}}$  est mise à 1. Ce qui signifie que le premier objet libre trouvé par une fourmi appartenant à un nid vide est systématiquement pris.

- si l'objet  $\mathcal{O}_i$  appartient déjà à un nid  $\mathcal{N}_j$ , la fourmi va déterminer s'il est préférable de la prendre ou de la laisser dans son nid en déterminant une similarité entre  $\mathcal{O}_i$  et le nid  $\mathcal{N}_k$  par :

$$g(\mathcal{O}_i, \mathcal{N}_k) = \left( \frac{f^{\text{pickup}}(\mathcal{O}_i, \mathcal{N}_j)}{f^{\text{pickup}}(\mathcal{O}_i, \mathcal{N}_k) |\mathcal{N}_j|} |\mathcal{N}_k| + |\mathcal{N}_j| \right) + \varepsilon \quad (6.3)$$

Où  $\varepsilon$  est une petite valeur arbitraire. Si  $g(\mathcal{O}_i, \mathcal{N}_k) < 1$  alors  $\mathcal{O}_i$  est considéré être plus similaire au nid  $\mathcal{N}_k$  qu'à son propre nid, de ce fait, la fourmi prend l'objet  $\mathcal{O}_i$ . Le rôle de cette fonction est d'encourager la formation rapide de nid contenant le même type d'objet.

**Le comportement des fourmis pour le dépôt des objets.** Lorsqu'une fourmi vient juste de prendre un objet  $\mathcal{O}_i$ , elle retourne dans son nid  $\mathcal{N}_j$  pour déposer l'objet dans une cellule libre du voisinage  $\mathcal{E}(\mathcal{N}_j)$  du nid. Pour chaque cellule  $c$  de  $\mathcal{E}(\mathcal{N}_j)$  la probabilité de déposer l'objet  $\mathcal{O}_i$  est donnée par :

$$P_{\text{drop}}(\mathcal{O}_i, c) = \left( \frac{f^{\text{drop}}(\mathcal{O}_i, c)}{k_d + f^{\text{drop}}(\mathcal{O}_i, c)} \right)^2 \quad (6.4)$$

De façon similaire à l'équation 6.1,  $k_d$  est une constante donnée comme paramètre de l'algorithme. La fonction  $f^{\text{drop}}$  est une de fonction de densité calculée comme suit :

$$f^{\text{drop}}(\mathcal{O}_i, c) = \max \left\{ \frac{1}{|v(c)|} \sum_{\mathcal{O}_k \in v(c)} 1 - \frac{d(\mathcal{O}_i, \mathcal{O}_k)}{\alpha}; 0 \right\} \quad (6.5)$$

Nous rappelons que  $\alpha$  est le même paramètre que dans l'équation 6.2 et  $v(c)$  est l'ensemble des objets qui sont dans le voisinage direct de la cellule  $c$ . Cependant, si  $v(c) = \emptyset$  alors  $f^{\text{drop}}(\mathcal{O}_i, c) = 1$  (la probabilité de déposer un objet dans une cellule qui n'en a aucun dans son voisinage est de 1).

Pour résumer les deux comportements de prise et dépôt d'une fourmi, nous pouvons dire qu'à chaque fois qu'une fourmi prend un objet, elle le ramène directement dans son nid, le dépose dans le voisinage du nid et retourne directement à la cellule où elle a trouvé l'objet. Ceci est exactement ce que font les fourmi *Pachycondyla apicalis* lorsqu'elles capturent une proie. Ces deux comportements (prise et dépôt d'objets) sont détaillés dans l'algorithme 9.

**Le comportement de recrutement.** En plus des comportements de recherche de nourriture des fourmis, il est possible pour une fourmi d'en recruter une autre. Le principal objectif de ce comportement est, pour la population de fourmis, d'auto-adapter la taille de chaque nid à celle du cluster. Par conséquent, avec ce mécanisme, il est possible qu'un nid se vide de ses fourmis si ces dernières ont été attirées par un autre nid et qu'il disparaisse de la grille.

**La construction de nouveaux nids.** Comme mentionné précédemment, le recrutement permettrait à certains nids de disparaître, de ce fait il serait intéressant d'introduire un comportement qui contrebalance cela pour créer des nids lorsque certaines conditions sont remplies.

Ces deux derniers comportements sont intéressants lorsque nous nous focalisons sur la capacité de API à trouver un nombre adapté de classes dans les données. De ce fait, elles n'ont pas été implémentées dans la version que nous allons présenter dans la section suivante.

## 6.3 Expérimentations

Dans un premier temps nous avons besoin de déterminer si API est capable de produire des résultats valides pour une classification de données sur des bases de données classiques ainsi que des données générées aléatoirement. Dans notre cas, nous allons présenter des résultats pour la version semi-supervisée et décrire les tentatives pour rendre API non-supervisée. Nous n'avons pas envisagé une classification supervisée avec API.

Ensuite, les résultats fournis par la méthode sont évalués. Cette évaluation est à faire suivant certains types d'attributs et des indices. Nous avons présenté dans le chapitre 5. Ces indices, bien qu'ils servent à évaluer un partitionnement, sont à adapter au domaine d'application. Mais dans notre cas (étude de API pour la classification), il s'agit d'une étude générale. Une des raisons pour lesquelles nous avons été exhaustif sur ces indices tient au fait que dans le domaine de la classification par fourmis, ce sont les mêmes indices qui ont tendance à être utilisés. Il nous a paru important d'inscrire notre étude dans le cadre plus générale de la classification en implémentant ces indices dans notre méthode.

Un autre point important concerne les paramètres de la méthode. Nous avons rapporté que le choix des paramètres des algorithmes de fourmis basés sur la méthode de Lumer et Faieta n'est pas toujours justifié. Nous allons évaluer notre méthode avec des valeurs différentes et tenter de comprendre dans quelle mesure ces paramètres influencent le modèle. Enfin, nous allons présenter une comparaison de nos résultats avec un certain nombre d'algorithmes de classification, en commençant par des algorithmes de fourmis et d'autres

**Algorithme 9** : API pour la classification de données

Initialisation : les objets et les nids sont dispersés aléatoirement sur la grille  $\mathcal{G}$ , les fourmis sont sur la même cellule que leur nid respectif.

```

pour  $T = 1$  à  $T_{\max}$  faire
  pour chaque nid  $\mathcal{N}_i \in \mathcal{N}$  faire
    pour chaque fourmi  $\mathcal{A}_j$  du nid  $\mathcal{N}_i$  faire
      si  $\mathcal{A}_j$  ne porte pas un objet alors
        si l'objet  $\mathcal{O}_k$  est sur la même cellule que  $\mathcal{A}_j$  alors
          si  $\mathcal{O}_k$  n'appartient à aucun nid alors
            la fourmi  $\mathcal{A}_j$  prend l'objet  $\mathcal{O}_k$ 
          fin si
          sinon
            si l'objet  $\mathcal{O}_k \in \mathcal{N}_\ell$  avec  $\ell \neq i$  alors
              si  $g(\mathcal{O}_k, \mathcal{N}_i) < 1$  alors
                 $\mathcal{O}_k$  est pris par la fourmi  $\mathcal{A}_j$  avec la probabilité
                 $P_{\text{pickup}}(\mathcal{O}_k, \mathcal{N}_i)$  (voir la formule 6.2)
              fin si
            fin si
          fin si
        fin si
        {Il n'y a aucun objet sur la cellule de la fourmi}
        La fourmi effectue un mouvement aléatoire dans son voisinage de Von
        Neuman
      fin si
    fin si
  sinon
     $\mathcal{A}_j$  mémorise sa position courante, retourne dans son nid  $\mathcal{N}_i$  et dépose
    l'objet  $\mathcal{O}_k$  dans  $\mathcal{E}(\mathcal{N}_i)$  selon la probabilité  $P_{\text{drop}}(\mathcal{O}_k, c)$  avec  $c \in \mathcal{E}(\mathcal{N}_i)$  (Voir
    formule 6.4)
     $\mathcal{A}_j$  retourne dans la cellule mémorisée
  fin si
fin pour
fin pour

```

Retourne : La position de chaque objet sur la grille et son nid correspondant

algorithmes provenant d'autres paradigmes.

### 6.3.1 Description des données

#### Les données générées aléatoirement

La base de données aléatoires a été générée par des gaussiennes et des lois uniformes dans le cas de art7 et art8. Nous allons utiliser, ces bases sont rassemblées dans la table 6.1.

Ces diverses bases sont illustrées dans l'annexe.

#### Les bases de données Machine Learning

**IRIS** est une base de données de 150 objets, dont chacun est décrit par 4 attributs de valeurs numériques et il y a trois classes initiales. Les attributs sont numériques.

**WINE** est constitué de 178 objets, dont chacun est décrit par 13 attributs numériques et ayant 3 classes. Les attributs sont numériques.

**Traitement de la distance** Comme nous l'avons vu dans le chapitre 5, la détermination de la distance est adaptée à chaque type d'attributs. Puis nous allons normaliser la distance.

### 6.3.2 Les fonctions d'évaluation

**F-Measure** est définie comme une moyenne harmonique de la précision ( $P$ ) et du rappel ( $R$ ) par [Sasaki, 2007] :

$$F = \frac{2}{\frac{1}{P} + \frac{1}{R}} = \frac{2PR}{P + R} \quad (6.6)$$

La précision et le rappel sont deux notions importantes qui ont été initialement utilisées en recherche de documents. Ils permettent de répondre à la question de savoir si une réponse à une requête retourne toute ou partie des documents recherchés, si ces documents sont importants et les autres documents retournés par la réponse mais qui ne sont pas liés à la requête. Plus la précision augmente et plus le rappel diminue (respectivement pour le rappel).

En classification de données, la précision et le rappel sont déterminés à l'aide d'une matrice de confusion qui est illustrée par la table 6.2. Soit une base de données composées de deux classes ( $c_i, i = 1..2$ ) et les classes ( $c'_j, j = 1..2$ ) trouvées par une méthode de classification. Les nombres  $n_{ij}$  correspondent aux cardinalités des classes et signifient le nombre d'objets de la classes  $c_i$  placés dans la classe  $c'_j$ .

La précision est déterminée par :

$$P = \frac{n_{11}}{n_{11} + n_{12}} \quad (6.7)$$

Le rappel est déterminé par :

$$R = \frac{n_{11}}{n_{11} + n_{21}} \quad (6.8)$$

### 6.3. EXPÉRIMENTATIONS

Nom	M	K	$ c_i $	répartition des points
art1	2	4	100	$(\mathcal{N}(0.2, 0.2), \mathcal{N}(0.2, 0.2))$
				$(\mathcal{N}(0.2, 0.2), \mathcal{N}(0.8, 0.2))$
				$(\mathcal{N}(0.8, 0.2), \mathcal{N}(0.2, 0.2))$
				$(\mathcal{N}(0.8, 0.2), \mathcal{N}(0.8, 0.2))$
art2	2	2	500	$(\mathcal{N}(0.2, 0.2), \mathcal{N}(0.2, 0.2))$
			500	$(\mathcal{N}(0.8, 0.2), \mathcal{N}(0.8, 0.2))$
art3	2	4	500	$(\mathcal{N}(0.2, 0.2), \mathcal{N}(0.2, 0.2))$
			50	$(\mathcal{N}(0.2, 0.2), \mathcal{N}(0.8, 0.2))$
			500	$(\mathcal{N}(0.8, 0.2), \mathcal{N}(0.2, 0.2))$
			50	$(\mathcal{N}(0.8, 0.2), \mathcal{N}(0.8, 0.2))$
art4	2	2	100	$(\mathcal{U}[-1, 1], \mathcal{U}[-10, 10])$
			100	$(\mathcal{U}[2, 3], \mathcal{U}[-10, 10])$
art5	2	9	100	$(\mathcal{N}(0.2, 0.2), \mathcal{N}(0.2, 0.2))$
				$(\mathcal{N}(0.8, 0.2), \mathcal{N}(0.2, 0.2))$
				$(\mathcal{N}(1.4, 0.2), \mathcal{N}(0.2, 0.2))$
			$\vdots$	$\vdots$
			100	$(\mathcal{N}(1.4, 0.2), \mathcal{N}(1.4, 0.2))$
art6	8	4	100	$(\mathcal{N}(0.2, 0.2), \dots, \mathcal{N}(0.2, 0.2))$
				$(\mathcal{N}(0.2, 0.2), \mathcal{N}(0.8, 0.2), \dots, \mathcal{N}(0.2, 0.2), \mathcal{N}(0.8, 0.2))$
				$(\mathcal{N}(0.8, 0.2), \mathcal{N}(0.2, 0.2), \dots, \mathcal{N}(0.8, 0.2), \mathcal{N}(0.2, 0.2))$
				$(\mathcal{N}(0.8, 0.2), \dots, \mathcal{N}(0.8, 0.2))$
art7	2	1		$(\mathcal{U}[0, 1], \mathcal{U}[0, 1])$
art8	2	1	1000	$(\mathcal{U}[0, 1], \mathcal{U}[0, 1])$
art9	4	3	200	$(\mathcal{N}(0.0, 0.2), \mathcal{N}(2.0, 0.5), \mathcal{N}(1.0, 0.3), \mathcal{N}(2.0, 0.7))$
				$(\mathcal{N}(0.0, 0.4), \mathcal{N}(1.0, 0.3), \mathcal{N}(4.0, 0.2), \mathcal{N}(1.0, 0.3))$
				$(\mathcal{N}(2.0, 0.3), \mathcal{N}(1.0, 0.4), \mathcal{N}(2.0, 0.9), \mathcal{N}(0.0, 0.4))$
art10	3	2	1000	$(\mathcal{N}(0.0, 0.3), \mathcal{N}(1.0, 0.1), \mathcal{N}(0.0, 1.2))$
				$(\mathcal{N}(2.0, 0.5), \mathcal{N}(2.0, 0.4), \mathcal{N}(1.0, 0.7))$

TABLE 6.1 – Caractéristiques des données aléatoires. M est le nombre d’attributs, K le nombre de classes et  $|c_i|$  le nombre d’objets dans chaque classe [Monmarché *et al.*, 2001].

		Classes trouvées par la méthode	
		$c_1$	$c_2$
Classes réelles	$c_1$	$n_{11}$	$n_{12}$
	$c_2$	$n_{21}$	$n_{22}$

TABLE 6.2 – Un exemple de matrice de confusion

Une généralisation [Chinchor et Sundheim, 1993] de l'équation 6.6 est donnée :

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \quad (0 \leq \beta \leq \infty) \quad (6.9)$$

Où  $\beta$  est un paramètre qui contrôle une balance entre  $P$  et  $R$ . Ainsi, l'équation donnée par 6.6, correspond à  $F_1$ .  $\beta$  permet d'orienter la signification de  $F_\beta$  soit vers  $R$  (lorsque  $\beta > 1$ ) et vers  $P$  ( $\beta < 1$ ).

Une autre version de F-mesure est exprimée en utilisant le paramètre  $\alpha$  :

$$F_\alpha = \frac{1}{\alpha \frac{1}{P} + (1 - \alpha) \frac{1}{R}} \quad (0 \leq \alpha \leq 1) \quad (6.10)$$

Avec  $\alpha = \frac{1}{\beta^2 + 1}$ . Le choix de la valeur de  $\alpha$  a été démontré par Van Risbergen qui a introduit la notion de F-mesure [C. J. van Rijsbergen, 1979] pour l'évaluation de l'extraction d'information.

**Erreur de classification** Soient deux objets  $i$  et  $j$  et les classes  $c$  d'origine et  $c'$  trouvée par l'algorithme.

L'erreur de classification,  $E_c$ , telle qu'elle a été utilisée dans [Monmarché *et al.*, 2001] :

$$E_c = \frac{2}{N(N-1)} \sum_{(i,j) \in \{1, \dots, N\}^2, i < j} e_{ij} \quad (6.11)$$

où :

$$e_{ij} = \begin{cases} 0 & \text{si } (c(i) = c(j) \wedge c'(i) = c'(j)) \vee (c(i) \neq c(j) \wedge c'(i) \neq c'(j)) \\ 1 & \text{sinon} \end{cases} \quad (6.12)$$

Cette erreur se base sur l'agrément et le désaccord, présentés dans le chapitre 5, pour calculer une erreur de classification. Si les paires sont bien classées l'erreur est nulle, sinon est elle de 1.

**Indice de Dunn** L'indice de Dunn est déterminé de la façon suivante [Dunn, 1974] :

$$D = \frac{\min_{1 \leq i \leq k} \{ \min_{1 \leq j \leq k} \{ \text{dist}(c_i, c_j) \} \}}{\max_{1 \leq l \leq k} \{ \text{diam}(c_l) \}} \quad (6.13)$$

Où :

$k$  est le nombre de classe. Et :

$$\text{diam}(c_l) = \max_{i,j \in c_l} \{d(i, j)\} \quad (6.14)$$

$$\text{dist}(c_i, c_j) = \min_{p \in c_i, q \in c_j, p \neq q} \{d(p, q)\} \quad (6.15)$$

Où :

$d(., .)$  est la distance euclidienne.

L'indice de Dunn est à maximiser ce qui veut dire qu'une valeur élevée indique que les classes ont été bien séparées.

### 6.3. EXPÉRIMENTATIONS

$\alpha = 0.1$	Base de données									
	art1	art2	art3	art4	art5	art6	art7	art8	art9	art10
Classification Error	0.39 (0.21)	0.49 (0.001)	0.46 (0.005)	0.49 (0.005)	0.20 (0.005)	0.41 (0.046)	0.002 (0.021)	0.108 (0.08)	0.45 (0.01)	0.49 (0.001)
F-Measure	0.013 (0.079)	0.21 (0.145)	0.18 (0.092)	0.23 (0.16)	0.76 (0.043)	0.13 (0.09)	0.402 (0.25)	0.39 (0.26)	0.171 (0.102)	0.21 (0.133)
Intra Cluster Variance	0.92 (0.48)	8.82 (4.67)	4.95 (2.50)	2.75 (1.33)	0.25 (0.087)	0.23 (0.099)	0.34 (0.18)	0.28 (0.14)	2.92 (1.56)	135.19 (73.85)
Dunn-Index	2.55 (1.65)	14.00 (6.40)	4.86 (2.37)	7.85(4.91)	0.78 ( 0.51)	1.165 (0.643)	7.27 10 <sup>8</sup> (0.0)	7.15 10 <sup>8</sup> (0.0)	6.25 (3.26)	25.23 (14.67)

TABLE 6.3 – Résultats obtenus (moyennes et écart-types) avec API et les bases artificielles pour  $\alpha = 0.1$  et 10 exécutions indépendantes.

$\alpha = 0.5$	Base de données									
	art1	art2	art3	art4	art5	art6	art7	art8	art9	art10
Classification Error	0.11 (0.002)	0.03 (0.003)	0.22 (0.009)	0.00 (0.004)	0.12 (0.007)	0.00 (0.00)	0.0 (0.0)	0.0 (0.0)	0.00 (0.0005)	0.00 (0.001)
F-Measure	0.87 (0.003)	0.98 (0.001)	0.76 (0.017)	0.99 (0.002)	0.60 (0.037)	0.99 (0.0008)	1.0 (0.0)	0.999 (0.0002)	0.99 (0.0004)	0.95 (0.0007)
Intra Cluster Variance	6.07 (0.023)	28.09 (0.075)	18.94 (0.46)	9.88 (0.04)	7.74 (0.348)	1.27 (0.003)	0.071 (0.0)	0.043 (0.0008)	7.95 (0.010)	64.42 (0.094)
Dunn-Index	28.7 (0.344)	161.34 (0.231)	48.60 (3.12)	28.99 (0.32)	4.72 (2.56)	9.25 (0.02)	7.25 10 <sup>8</sup> (0.0)	7.25 10 <sup>8</sup> (0.0)	34.77 (0.054)	194.13 (0.211)

TABLE 6.4 – Résultats obtenus (moyennes et écart-types) avec API et les bases artificielles pour  $\alpha = 0.5$  et 10 exécutions indépendantes.

**Variance intra classe** La variance intra classe est un critère à minimiser. Plus est petite, plus les classes trouvées sont compactes autour de leur centre de gravité. Nous renvoyons le lecteur au chapitre 5 pour plus de détails.

#### 6.3.3 Les résultats

##### Résultats préliminaires

Nos expériences ont été conduites, dans un premier temps, avec les valeurs suivantes pour les paramètres :  $\alpha = 0.5$ ,  $k_p = 0.1$ ,  $k_d = 0.15$ ,  $\varepsilon = 0.03$ , le nombre de fourmis est fixé à 10. Ces valeurs sont les plus couramment utilisées par les méthodes de classification par fourmis artificielles basées sur la méthode de Lumer et Faieta.

Les Figure 6.2 et 6.3 donnent l'évaluation de 3 exécutions de API (10,000 itérations) afin de montrer la tendance globale des indices de mesures de performance au cours du temps. Les résultats montrent que pour les deux bases de données, API fait évoluer les quatre indices dans le sens de bonnes performances. Nous voyons bien que la F-mesure croît, l'indice de Dunn croît et l'erreur de classification décroît. Quant à l'indice de variance inter-classe, nous l'avons modifié pour qu'une décroissance indique une classe compacte. Dans la suite, nous allons déterminer plus en détails pourquoi un tel comportement apparaît.

##### Étude des paramètres

**Le paramètre  $\alpha$**  Comme nous l'avons mentionné,  $\alpha$  a une importance capitale dans la forme des classes trouvées par API. Pour comprendre comment ce paramètre influence les résultats, nous pourrions étudier de façon analytique les deux fonctions  $f_{pickup}$  et  $f_{drop}$ . Cependant, dans cette section, nous avons choisi de donner des résultats expérimentaux de API en fonction de plusieurs valeurs de  $\alpha$  en conservant les mêmes valeurs pour les autres paramètres, et ce, pour les données artificielles. Toutes les expériences sont réalisées pour 1 million d'itérations (sauf indication contraire) et nous allons donner la moyenne et l'écart-type pour 10 exécutions indépendantes. Pour chaque base, nous conservons le même nombre de classe. Les tables 6.3-6.7 décrivent les résultats obtenus pour plusieurs valeurs de  $\alpha$ .

### 6.3. EXPÉRIMENTATIONS

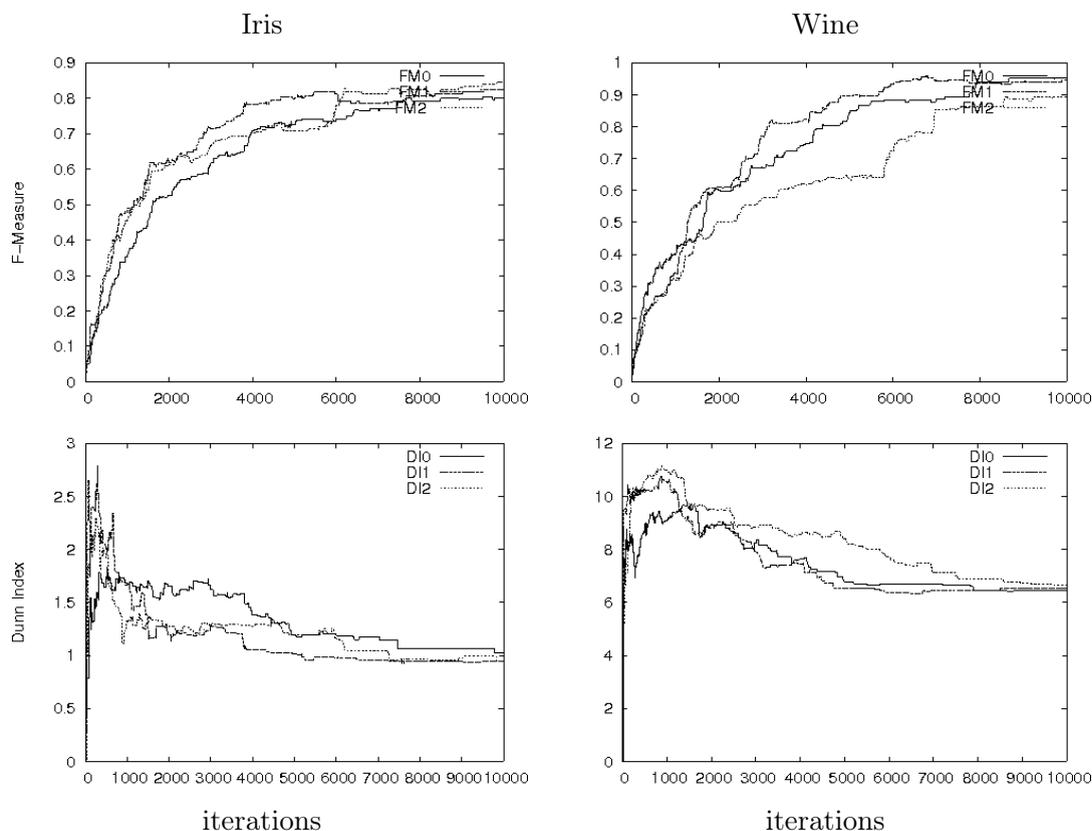


FIGURE 6.2 – Évolution des fonctions d'évaluations F-Measure et Dunn Index au cours de 10,000 itérations et 3 exécutions indépendantes sur les bases de données Iris et Wine.

$\alpha = 1$	Base de données									
	art1	art2	art3	art4	art5	art6	art7	art8	art9	art10
Classification Error	0.11 (0.006)	0.04 (0.003)	0.2 (0.017)	0.008 (0.007)	0.11 (0.008)	0.00 (0.001)	0.0 (0.000)	0.00 (0.001)	0.00 (0.0005)	0.01 (0.002)
F-Measure	0.87 (0.008)	0.97 (0.001)	0.73 (0.043)	0.99 (0.003)	0.65 (0.046)	0.99 (0.001)	1.0 (0.000)	0.99 (0.0002)	0.99 (0.0004)	0.99 (0.001)
Intra Cluster Variance	6.04 (0.039)	28.06 (0.123)	18.19 (1.238)	9.90 (0.021)	8.47 (0.260)	1.26 (0.004)	0.07 (0.000)	0.04 (0.001)	7.95 (0.0101)	64.40 (0.141)
Dunn-Index	28.06 (0.693)	161.19 (0.407)	44.58 (4.908)	29.05 (0.284)	7.04 (2.410)	9.22 (0.055)	$7.275 \cdot 10^8$	$7.15 \cdot 10^8$	34.77 (0.049)	194.43 (0.296)

TABLE 6.5 – Résultats obtenus (moyennes et écart-types) avec API et les bases artificielles pour  $\alpha = 1$  et 10 exécutions indépendantes.

$\alpha = 10$	Base de données									
	art1	art2	art3	art4	art5	art6	art7	art8	art9	art10
Classification Error	0.43 (0.048)	0.50 (0.0005)	0.46 (0.007)	0.50 (0.001)	0.21 (0.010)	0.41 (0.032)	0.00 (0.008)	0.00 (0.001)	0.48 (0.028)	0.49 (0.0005)
F-Measure	0.33 (0.019)	0.54 (0.0336)	0.39 (0.041)	0.59 (0.039)	0.16 (0.005)	0.32 (0.022)	0.98 (0.002)	0.99 (0.0002)	0.41 (0.002)	0.53 (0.0211)
Intra Cluster Variance	-	1.75 (0.2238)	1.16 (0.222)	0.56 (0.141)	0.13 (0.040)	0.06 (0.010)	0.07 (0.003)	0.04 (0.002)	0.66 (0.116)	25.67 (1.170)
Dunn-Index	0.75 (0.390)	6.37 (5.220)	1.83 (0.724)	2.21 (1.287)	0.28 (0.109)	0.38 (0.116)	$7.27 \cdot 10^8$	$7.15 (0.000) \cdot 10^8$	1.52 (0.544)	19.16 (12.591)

TABLE 6.6 – Résultats obtenus (moyennes et écart-types) avec API et les bases artificielles pour  $\alpha = 10$  et 10 exécutions indépendantes.

### 6.3. EXPÉRIMENTATIONS

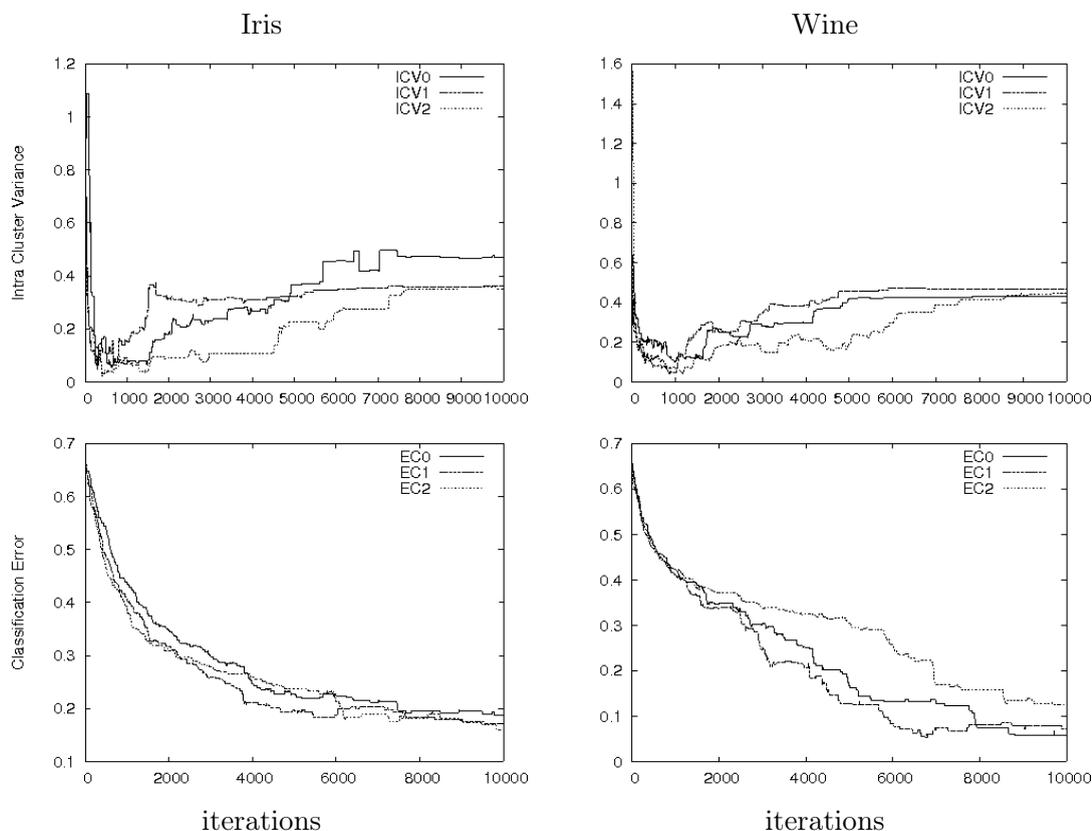


FIGURE 6.3 – Évolution des fonctions d'évaluations ICV (Variance Intra Classe) et EC (Erreur de classification) au cours de 10,000 itérations et 3 exécutions indépendantes sur les bases de données Iris et Wine.

$\alpha = 100$	Base de données									
	art1	art2	art3	art4	art5	art6	art7	art8	art9	art10
Classification Error	0.46 (0.067)	0.49 (0.0009)	0.46 (0.005)	0.50 (0.002)	0.21 (0.008)	0.44 (0.038)	0.00 (0.008)	0.001 (0.001)	0.46 (0.017)	0.49 (0.0005)
F-Measure	0.34 (0.020)	0.55 (0.0308)	0.38 (0.033)	0.65 (0.034)	0.16 (0.006)	0.33 (0.022)	0.99 (0.002)	0.99 (0.0003)	0.39 (0.025)	0.53 (0.0203)
Dunn-Index	-	1.69 (0.1853)	1.10 (0.162)	-	0.13 (0.045)	-	0.07 (0.002)	0.04 (0.002)	0.59 (0.083)	25.52 (0.689)
Intra Cluster Variance	0.65 (0.412)	5.17 (4.3432)	2.01 (1.047)	4.61 (1.060)	0.31 (0.193)	0.42 (0.191)	$7.27 \cdot 10^8$ (0.000)	$7.15 \cdot 10^8$ (0.000)	1.61 (0.808)	17.50 (11.65)

TABLE 6.7 – Résultats obtenus (moyennes et écart-types) avec API et les bases artificielles pour  $\alpha = 100$  et 10 exécutions indépendantes.

art5 - $\alpha = 0.5$	Nombre d'itérations (en millions)			
	2	3	5	10
Classification Error	0.12 (0.014)	0.11 (0.016)	0.12 (0.006)	0.12 (0.007)
F-Measure	0.63 (0.062)	0.64 (0.058)	0.62 (0.027)	0.63 (0.033)
Dunn-Index	8.34 (0.272)	8.41 (0.491)	8.35 (0.565)	8.09 (0.376)
Intra Cluster Variance	5.41 (3.453)	5.60 (2.997)	4.79 (1.927)	4.82 (2.862)

TABLE 6.8 – Résultats obtenus (moyenne et écart-type) avec API et la base art5 lorsque  $\alpha = 0.5$  pour 10 exécutions indépendantes et un nombre variable d'itérations.

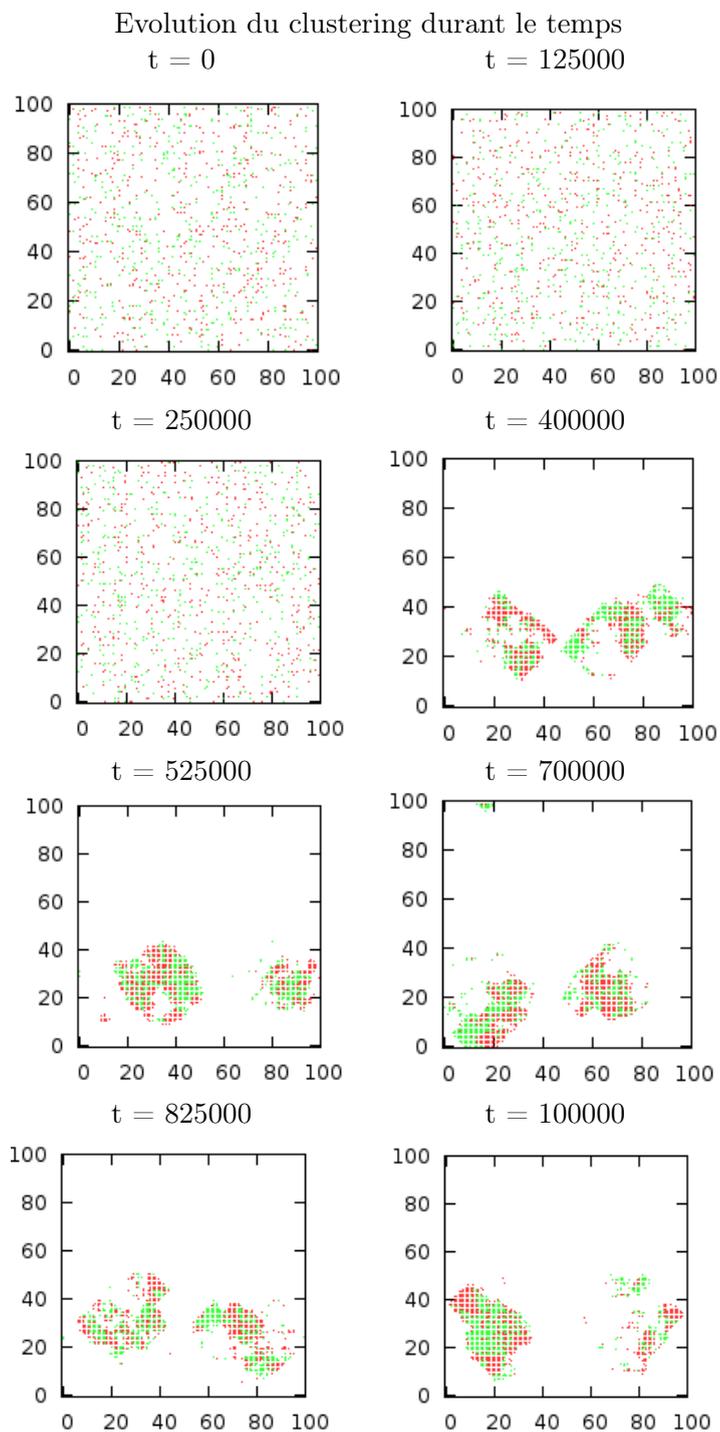


FIGURE 6.4 – Évolution de la répartition des données dans les classes trouvées par la méthode API pour la base art2 ( $\alpha = 0.5$ ,  $kp = 0.1$ ,  $k_d = 0.15$ ). De  $t = 0$  à  $t = 250000$  les classes sont formées sur la grille mais pas encore mise à jour pour l’affichage.

### 6.3. EXPÉRIMENTATIONS

---

art5 - $\alpha = 0.5$	Nombre de fourmis			
	9	90	450	900
Classification Error	0.12 (0.006)	0.12 (0.002)	0.12 (0.015)	0.12 (0.011)
F-Measure	0.59 (0.034)	0.61 (0.019)	0.64 (0.066)	0.62 (0.049)
Dunn-Index	8.4 (0.713)	8.27 (0.390)	8.41 (0.377)	8.28 (0.532)
Intra Cluster Variance	3.25 (1.936)	6.11 (2.943)	5.99 (4.143)	5.23 (2.838)

TABLE 6.9 – Résultats obtenus (moyenne et écart-type) avec API et la base art5 lorsque  $\alpha = 0.5$ , pour 10 exécutions indépendantes et un nombre variable de fourmis (900 correspond au nombre d'objet dans la base art5).

art6 - $\alpha = 0.5$	$k_p$						
	10	100	1000	-10	-100	-1000	-0.1
Classification Error	0.001 (0.001)	0.001 (0.0009)	0.001 (0.001)	0.001 (0.001)	0.000 (0.001)	0.001 (0.001)	0.001 (0.001)
F-Measure	0.998 (0.001)	0.998 (0.0009)	0.998 (0.001)	0.998 (0.001)	0.999 (0.001)	0.998 (0.001)	0.998 (0.001)
Dunn-Index	1.269 (0.004)	1.269 (0.0028)	1.268 (0.003)	1.268 (0.004)	1.271 (0.001)	1.268 (0.004)	1.267 (0.006)
Intra Cluster Variance	8.84 (1.044)	9.25 (0.0254)	92.4 (0.038)	9.14 (0.218)	9.260 (0.021)	9.04 (0.722)	8.68 (1.216)

TABLE 6.10 – Résultats obtenus avec API et la base art6 pour plusieurs valeurs de  $k_p$  et 10 exécutions indépendantes.

art6 - $\alpha = 0.5$	$k_d$						
	10	100	1000	-10	-100	-1000	-0.1
Classification Error	0.00 (0.000)	0.001 (0.001)	0.000 (0.001)	0.001 (0.001)	0.001 (0.001)	0.001 (0.001)	0.002 (0.001)
F-Measure	1.00 (0.000)	0.998 (0.001)	0.999 (0.001)	0.998 (0.001)	0.998 (0.001)	0.998 (0.001)	0.997 (0.001)
Dunn-Index	1.27 (0.000)	1.26 (0.005)	1.26 (0.005)	1.26 (0.005)	1.26 (0.004)	1.26 (0.003)	1.26 (0.005)
Intra Cluster Variance	9.27 (0.000)	8.65 (1.287)	8.94 (1.013)	8.72 (1.170)	8.49 (1.354)	9.22 (0.050)	8.95 (0.853)

TABLE 6.11 – Résultats obtenus avec API et la base art6 pour plusieurs valeurs de  $k_p$  et 10 exécutions indépendantes.

Le premier constat (table 6.3) est qu'une petite valeur de  $\alpha$  (0.1) a un impact plus important sur la densité. En effet, dans le cas où  $\alpha = 0.1$  la similarité est multipliée par 10 (voir les équations 6.2 et 6.5). Ce qui a comme effet d'empêcher la prise en compte d'objets qui sont proches. De ce fait, pour toutes les bases, sauf art7 et art8, les résultats trouvés sont mauvais. art7 et art8 ne contiennent qu'une seule classe : les résultats de la classification pour ces classes ne dépendront d'aucun des paramètres ( $\alpha, k_p, k_d$ ) sauf du nombre d'itérations. En effet, pour ces deux bases API est initialisé avec un nombre de classe à trouver égale à un. Ainsi, une classe sera trouvée mais si le nombre d'itérations n'est pas suffisante certaines données ne seront pas classées et ce sera la principale source d'erreur. L'indice de Dunn pour ces deux bases est très élevé (de l'ordre de  $10^8$ ), ce qui s'explique par le fait que la plus petite distance dans une même classe tend vers 0.

La valeur  $\alpha = 0.5$  donne de meilleurs résultats que  $\alpha = 0.1$ , et ceci pour toutes les bases. En effet, dans ce cas, l'écart dans la fonction de densité n'est pas très important que lorsque  $\alpha = 0.1$ . Rappelons que les distances sont normalisées entre 0 et 1 et il revient à multiplier la distance par 2 dans la fonction de densité pour  $\alpha = 0.5$ . Cependant, nous constatons qu'en fonction du nombre de classes de la base, API classe plus ou moins bien les données. C'est le cas des bases art1, art3 et art5. Pour la base art1 par exemple, nous avons une moyenne de 11% d'erreur de classification alors que cette valeur est quasiment nulle pour les autres bases. Ces résultats montrent qu'une valeur bonne pour les autres bases ne l'est pas forcément pour art1, art3 et art5. Les résultats fournis pour  $\alpha = 1$ , donnés dans le tableau 6.5, sont assez proches de celles  $\alpha = 0.5$  avec les mêmes constats pour les bases art1, art3 et art5.

Cependant, plus la valeur de  $\alpha$  augmente, et ceci pour toutes les bases, moins les résultats sont bons. C'est le cas notamment des valeurs de  $\alpha = 10$  et  $\alpha = 100$  (voir tableau 6.6 et 6.7). Puisque les distances sont normalisées, les données éloignées auront tendance à être rassemblées dans la même classe lorsque  $\alpha = 10$  et  $\alpha = 100$ . Ce fait est montré par les faibles valeurs de la variance intra-classe (voir tableau 6.6) lorsque la valeur de  $\alpha$  augmente.

Dans le cas de art5 (un nombre de classe important), on peut considérer les points suivants pour améliorer le résultats :

- le nombre d'itérations : est-ce que le nombre d'itérations est suffisant pour que la méthode converge ? Pourrait-on considérer un critère d'arrêt qui n'est pas fixé à l'avance mais déterminé par les fourmis ?
- le nombre de fourmis : est-ce que le nombre de fourmis pour un nombre d'itérations donné est suffisant ?
- le paramètre  $k_p$  : ce paramètre peut il être choisi de sorte qu'il assure une prise rapide des objets pour que la fourmi puisse faire plusieurs actions ?
- le paramètre  $k_d$  : la question précédente est identique ici, en considérant le dépôt.

La table 6.8 présente les résultats obtenus pour la base art5 avec  $\alpha = 0.5$  et différentes valeurs pour le nombre d'itérations. En nous basant sur ces résultats, nous pouvons conclure que le nombre d'itérations, pour une valeur de  $\alpha$  considérée bonne, n'a pas une très grande influence. Généralement, comme illustré par la figure 6.4, la méthode trouve les classes assez rapidement (même si, dans ce cas de figure, nous constatons un démarrage lent), et les fourmis ne déplacent quasiment plus d'objets. Dans ce cas, nous pouvons compter le nombre d'actions effectuées par les fourmis et arrêter API si des actions ne sont plus

entreprises.

Un dernier point concernant le nombre d'itérations est la façon dont chaque fourmi est choisie. Nous avons implémenté une exécution séquentielle. Ce qui signifie que c'est d'abord le premier nid qui est traité, puis la première fourmi, etc. On peut craindre que cette façon de faire influence le résultat. C'est pour cela que certains auteurs ont implémenté un tirage aléatoire des fourmis. Cependant, même si le premier nid semble être avantageux, ce fait est contrebalancé par le déplacement des objets à la condition que le nombre d'itérations soit suffisante pour que les fourmis trouvent tous les objets sur la grille.

Analysons maintenant le nombre de fourmis. Comme nous l'avons décrit, un même nombre est attribué à chaque nid. Lorsque nous disposons d'un même nombre d'objets par classe, aucune classe ne semble être avantageuse. Cependant, il n'est pas facile d'adapter le nombre de fourmis dès le départ à chaque nid. D'où la nécessité de mettre en place une méthode heuristique. Pour cette analyse, nous allons considérer toujours 1 millions d'itérations,  $\alpha = 0.5$ , et faire varier le nombre de fourmis. Il faut au minimum que le nombre de fourmis soit égal au nombre de nids dans les données, sinon certains nids n'auront pas de fourmis. La table 6.9 montre que la variation du nombre de fourmis n'impacte pas significativement les résultats. Ceci s'explique par le fait que la fonction de densité et aussi la fonction de dépôt permettent de bien classer les objets pour avoir des nids de forte densité. Ce qui empêche les fourmis de les prendre. Cependant, le nombre de fourmis ralentit le programme.

Pour étudier les valeurs de  $k_p$  et  $k_d$ , nous allons considérer, la base art6 qui est bien classée pour les valeurs de  $\alpha$  à 0.5 et 1. En suivant le même principe, c'est-à-dire,  $\alpha = 0.5$ , le nombre de fourmis est de 10 et le nombre d'itérations est de 1 million. Les résultats donnés par la table 6.10 et 6.11 montrent que lorsque les classes sont facilement séparables, l'influence de  $k_p$  n'est pas très importante.

C'est pour cette raison que certains auteurs [Handl et Meyer, 2007] ont proposé des fonctions de prise et de dépôt dépourvue de ce paramètre. Cependant nous les conservons pour éviter les divisions par zéro dans les fonctions de probabilité et peuvent être remplacés par des constantes.

Nous allons comparer cette version de API avec un certain nombre de méthodes de clustering.

### Étude comparative

La principale contribution de API pour le clustering par fourmis artificielles est l'introduction du concept de nid. Ainsi par rapport aux méthodes :

- de Julia Handl [Handl et Meyer, 2002, Handl et Meyer, 2007, Handl *et al.*, 2003];
- de Urszula Boryczka [Boryczka, 2008, Boryczka, 2009];

API n'a pas besoin d'une autre méthode pour construire les classes. En effet, Julia Handl (et les méthodes dérivées, notamment les travaux de Boryczka) utilise d'autres types d'algorithmes (des méthodes hiérarchiques ou des cartes de Kohonen) pour construire les classes après que les fourmis aient regroupé en tas sur la grille. Ainsi, l'algorithme de classification par fourmis présenté par les auteurs est aidé par la méthode de clustering. Au final, la méthode se complexifie et possède les inconvénients et les paramètres de la seconde méthode.

### 6.3. EXPÉRIMENTATIONS

	Iris dataset		Wine dataset	
	API	K-Means	API	K-Means
EC	0.223(0.000)	0.223 (0.000)	0.31 (0.004)	0.315 (0.009)
F-Measure	0.777 (0.000)	0.777 (0.000)	0.652 (0.012)	0.677 (0.016)
Dunn-Index	11.473 (0.040)	0.575 (0.000)	2.851 (0.303)	0.309 (0.004)
Intra Cluster Variance	2.76 (0.007)	0.0181 (0.000)	0.666 (0.008)	0.106 (0.0004)

TABLE 6.12 – Résultats obtenus avec API et K-means pour Iris et Wine pour  $k = 2$ .

	Iris dataset		Wine dataset	
	API	K-Means	API	K-Means
Classification Error	0.166 (0.008)	0.211 (0.019)	0.108 (0.006)	0.114 (0.019)
F-Measure	0.800 (0.008)	0.753 (0.029)	0.889 (0.008)	0.870 (0.037)
Dunn-Index	2.342 (0.517)	0.396 (0.089)	1.328 (0.162)	0.194 (0.023)
Intra Cluster Variance	1.915 (0.374)	0.019 (0.001)	0.345 (0.007)	0.162 (0.008)

TABLE 6.13 – Résultats obtenus avec API et K-means pour Iris et Wine pour  $k = 4$ .

Cette remarque est aussi valable pour les méthodes ayant hybridé le comportement des fourmis avec d'autres algorithmes de clustering par exemple Antclass [Monmarché *et al.*, 2001]. Cependant, l'hybridation sert aussi à compenser les défauts d'une méthode.

L'inconvénient du concept de nid est qu'il faut en fixer le nombre à l'avance pour une version semi-supervisée. Cela rajoute aussi pour la version non supervisée une difficulté supplémentaire.

Un avantage du concept de nid est la simplification de l'algorithme. La majorité des modifications introduites depuis Lumer et Faieta ont pour but de permettre une convergence rapide et de permettre aux fourmis de trouver le nombre de classes (notion diffuse car, elle est confondue avec les tas). De ce fait, certains auteurs ont introduit le kernel gaussien. C'est le cas de Gilbert K. Paterson dans [Peterson *et al.*, 2008]. Le but de cette fonction kernel c'est d'optimiser la gestion de la mémoire des objets rencontrés. Dans API, nous n'avons pas besoin d'une mémoire ni de fonction complexe de gestion de la mémoire, car la fourmi retourne systématiquement à la dernière place où elle a trouvé un objet.

Nous allons comparer nos résultats à k-means et ACA sur les bases Iris et Wine. La Table 6.12 (*resp.* Table 6.13) présente la moyenne et l'écart-type (obtenus pour 10 exécutions indépendantes) de la fonction d'évaluation pour API (10.000 itérations) et K-means lorsque le nombre de clusters est  $k = 2$  (*resp.*  $k = 4$ ).

La Table 6.14 et 6.15 présentent la comparaison des résultats de API, K-means et ACA. API et ACA sont exécutés (10 exécutions indépendantes) durant 1.000.000 d'itérations avec 10 fourmis. La table donne aussi la moyenne et l'écart-type obtenus avec Iris et Wine pour les quatre indices. Selon les résultats obtenus avec les 3 mesures, les résultats de API sont proches de ceux de K-means, et améliore sensiblement les résultats de ACA. K-means a de meilleurs résultats que API et ACA sur la base Wine.

<b>Iris dataset</b>	<b>API</b>	<b>K-Means</b>	<b>ACA</b>
Classification Error	0.123 (0.004)	0.165 (0.009)	0.230 (0.053)
F-Measure	0.89 (0.006)	0.838 (0.015)	0.773 (0.022)
Dunn-Index	5.04 (0.106)	0.411 (0.068)	2.120 (0.628)
Intra Cluster Variance	2.37 (0.089)	0.018 (0.0004)	4.213 (1.609)

TABLE 6.14 – Résultats obtenus avec API et K-means, ACA pour Iris.

<b>Wine dataset</b>	<b>API</b>	<b>K-Means</b>	<b>ACA</b>
Classification Error	0.072 (0.007)	0.053 (0.006)	0.142 (0.030)
F-Measure	0.943 (0.006)	0.96 (0.004)	0.855 (0.023)
Dunn-Index	2.233 (0.177)	0.293 (0.005)	1.384 (0.101)
Intra Cluster Variance	0.460 (0.133)	0.123 (0.035)	8.521 (0.991)

TABLE 6.15 – Résultats obtenus avec API et K-means, ACA pour Iris

## 6.4 Vers une version non supervisée de API

Nous tenons pour ce chapitre à donner un certain nombre de travaux que nous avons entrepris. Pour ce faire nous allons décrire les points suivants :

- le recrutement : nous avons mise au point une méthode permettant un recrutement de fourmis. Elle est la suivante : Soit deux fourmis A et B. La fourmi A recrute la fourmis B si :
  1. B ne fait rien. Une fourmi B ne fait rien lorsque B appartient à un nid vide et B ne porte pas d'objet ;
  2. B fait la même chose que A. Pour déterminer ce cas, nous avons besoin d'une fonction qui mesure la proximité des tâches. Nous avons envisagé, pour ce cas, l'usage des fonctions de similarité utilisé par les méthodes hiérarchiques pour fusionner deux classes. Dans ce cas, A ne recrute que B.
  3. Un autre cas de figure est lorsque les deux nids sont assez proches. La fourmis peut recruter toutes les autres fourmis appartenant au nids en en recrutant un seul qui par *tendem running* recrutera les autres fourmis de son nid.
- API pour la classification non supervisée : il s'agit ici de déterminer le nombre de classe automatiquement. Ce comportement est à combiner avec le recrutement qui fait baisser le nombre de nid. Pour créer un nid nous avons envisagé les cas suivants :
  1. utiliser les critères d'information (Akaike, Bayésien, etc) ou Gap Statistic pour déterminer un nombre de classes de départ. Ce comportement est encore semi-supervisée ;
  2. une règle simple est que lorsque la fourmi n'a pas de nid, elle se crée son propre nid. Il y aura autant de classes que de fourmis. C'est le recrutement qui dans ce cas aussi est envisagé pour diminuer le nombre de classes ;
  3. tirer aléatoirement un nombre de nid, et le faire baisser par le recrutement ou aussi l'augmenter par la création de nouveau nid.

- les bases de Machine Learning. Il reste en effet, à étudier la méthode sur ces bases comme nous l'avons annoncé et avec les fonctions de similarité appropriées aux données.

Nous avons implémenté et testé toutes les propositions faites ici, sauf les critères d'informations. Dans notre implémentation, le recrutement utilise les fonctions de similarité données dans la section 5.5 du chapitre 5 traitant de la classification hiérarchique. Cependant, les résultats ne sont pas assez bons pour publication.

## 6.5 Les fourmis *P. Apicalis* pour la localisation interne

Pour appliquer API à la localisation *indoor* : deux cas de figure sont à distinguer :

1. considérer le problème de localisation comme un problème de classification.  
Généralement en localisation *indoor*, ce sont les méthodes supervisées qui sont les plus exploitées dans le cas du fingerprinting notamment. Nous pouvons considérer deux cas de figure :
  - (a) mettre au point une version supervisée de API et l'appliquer au fingerprint ;
  - (b) le problème de positionnement *indoor* est modélisé comme suit : en fonction des données collectées, un nœud va établir un clustering. Chaque classe pourra correspondre à une position. Cette dernière correspondra soit à une chambre ou une partie d'une chambre. Nous voyons que ce positionnement n'est pas très précis. Par exemple, une approche similaire a été développée dans [Dakkak *et al.*, 2011], les auteurs ont appliqué les filtres à particules (Particle Swarm Optimization) au positionnement interne. Il s'agissait de déterminer par clustering la position d'un nœud mobile en fonction des nœuds fixes et en se basant sur des données obtenues par temps d'arrivée et angle d'arrivée. Les positions trouvées sont exprimées en coordonnées de deux ou trois dimensions.
2. dans ce cas de figure la modélisation du problème est le suivant : les nœuds fixes et mobiles (dont les positions sont inconnues) sont considérés comme des fourmis. Il nous faudra adapter le comportement des fourmis *P. apicalis* pour générer un algorithme distribué permettant de positionner les nœuds. Il faudra prendre en compte le positionnement : centralisé ou distribué, le type de représentation de l'environnement *indoor* et tous les autres points que nous avons abordés dans le premier chapitre sur les réseaux de capteurs, notamment leur modélisation.

## 6.6 Conclusion

Ce chapitre a présenté nos travaux pour l'adaptation de la méthode API sur la classification automatique de données. Les résultats que nous avons obtenus sont encourageants à plus d'un titre. En effet, nous pouvons affirmer que sous certaines conditions, la version que nous avons présentée dans ce chapitre trouve de façon satisfaisante les classes comparativement à certaine méthode de fourmis. Bien que la question de la localisation par cette méthode n'ait pas été abordée de façon approfondie, la méthode permettra une fois que le

## 6.6. CONCLUSION

---

problème de la classification ait été modélisé en problème de clustering de déterminer la position.

# Conclusion générale

La première question que nous avons posée, après avoir analysé le sujet de la thèse est : comment, et pourquoi, positionner une personne handicapée dans un environnement intérieur ? Cette analyse nous a conduit à un découpage en trois sous-problèmes :

1. le handicap ;
2. le positionnement ;
3. les environnements intérieurs.

Le handicap est, en résumé, une situation temporaire ou permanente dans laquelle une personne ne peut pas jouir de toutes ses facultés physiques, psychiques et sensorielles. Comme nous l'avons vu, dans chaque pays, les politiques et les moyens de prise en charge du handicap sont très divers. Certains pays sont quand même mieux dotés, du moins sur le plan législatif, pour accorder les mêmes chances aux personnes handicapées de mieux contribuer à la société et s'épanouir .

Les systèmes de positionnement sont des outils permettant, entre autre, d'aider les personnes handicapées dans leur vie de tous les jours. Ils constituent la couche basse des applications dites « basées sur la position » ou « basées sur le contexte ». En effet, une position est une information brute à partir de laquelle plusieurs types d'applications peuvent être construits. Le contexte est plus large que la position et inclut cette dernière aussi bien que la santé, les conditions atmosphériques, etc.

Les environnements intérieurs sont, par extension, tout milieu dans lequel les systèmes de positionnement extérieurs (comme le GPS) ne fonctionnent pas bien. En effet, ces systèmes voient leurs technologies s'améliorer de plus en plus et atteindre divers types de précisions, par exemple, des précisions différentes pour les applications civiles ou militaires. Par contre, ces technologies fonctionnent mal dans les environnements intérieurs. La principale raison est la difficulté qu'ont les ondes électromagnétiques à se propager dans ces milieux. Précisons que d'autres technologies sont utilisées pour la localisation *in-door*, mais en raison de leur coût et de la difficulté à les exploiter facilement, les ondes électromagnétiques leur sont préférées.

Après l'étude de ces divers domaines, il ressort qu'une solution technologique n'est pas suffisante dans la gestion du handicap. Une prise en compte efficace du handicap doit s'effectuer avec toutes les parties impliquées dans la gestion du handicap. Ceci dit, du côté du concepteur d'un système de positionnement, le handicap doit être modélisé, entre autres, en termes de besoins de l'utilisateur, de précisions requises par le type de handicap, de la conception du nœud pour éviter le rejet et la préservation de la vie privée.

Le point de vue qui a été le notre pour apporter des réponses à la question posée, a été dans un premier temps de considérer qu'il faut établir une distinction entre algorithmes de positionnement et technologies de positionnement. Nous sommes arrivés à cette conclusion après l'étude des réseaux de capteurs. En effet, au tout début de nos travaux, nous étions orientés directement sur les réseaux de capteurs. Ces réseaux sont constitués de nœuds ayant de faibles capacités en terme de calcul, de transmission et d'énergie. Plusieurs auteurs, dans l'état de l'art que nous avons effectué, liaient la possibilité d'utiliser un algorithme et ses résultats uniquement aux réseaux de capteurs. Cependant lorsque nous avons considéré d'autres types de réseaux, comme les réseaux WiFi et même les systèmes satellitaires et les réseaux cellulaires, nous avons trouvé que les mêmes algorithmes étaient utilisés. Ainsi, par la suite, nous avons conclu qu'il était possible d'utiliser aussi bien une méthode géométrique simple, qu'une méthode bayésienne et des algorithmes bio-inspirés pour résoudre un problème de positionnement *indoor* dans tout type de réseau à la condition que le réseau fournisse le type de données requis par la méthode (ce qui est en théorie le cas pour la majorité des normes de transmission exploitant les ondes électromagnétiques ou mécaniques même si des limitations sont introduites par les équipements fabriquées par les constructeurs). A partir de là, nous avons établi une extension de la notion de réseaux de capteurs et aussi des modèles pour la génération de données pour une technologie de transmission particulière.

Pour la première contribution de cette thèse, nous avons considéré que, dans certaines circonstances, une simple méthode géométrique suffit pour fournir une position remplissant les exigences en terme de précision. Nous avons proposé un système de positionnement basé sur les réseaux de capteurs (au sens strict, c'est à dire sans l'extension) utilisant la norme de transmission Zigbee et des algorithmes géométriques. Nos expériences ont été conçues afin d'étudier ces circonstances : déploiement des nœuds de références, conditions de propagation et exigences en termes de précision.

La seconde contribution est venue avec le constat qu'une personne est amenée à naviguer dans plusieurs milieux différents (ayant des technologies différentes), nous avons proposé un système capable d'utiliser les divers réseaux (WiFi, Bluetooth, IEEE 802.15.4, Zigbee, RFID/NFC, GSM) pour localiser la personne. Ce système est basé sur un middleware qui a été conçu pour la transparence, l'interopérabilité et la facilité de développer des services basés sur la position. Sur le plan algorithmique, nous avons envisagé de tester toutes les méthodes de positionnement *indoor* que nous avons étudié.

Dans la suite de nos travaux, il s'agit d'étudier les apports possibles des algorithmes de fournis pour les réseaux de capteurs appliqués à la localisation de personnes handicapées. Nous avons montré que les algorithmes de fournis pouvaient contribuer au positionnement des personnes handicapées lorsqu'on transforme le problème de positionnement en problème de clustering ou d'optimisation. Nous avons mis au point une méthode de clustering (pouvant classifier plusieurs types de données) afin de l'appliquer au problème de positionnement. Il s'agit de la méthode API ou *pachycondylla apicallis*. Nous avons revu plusieurs catégories de méthodes de clustering dans la perspective de nous en inspirer et aussi de comparer les résultats de notre méthode. Les résultats que nous avons obtenus pour API en classification sont prometteurs.

Au moment de la rédaction de cette conclusion, le positionnement à l'intérieur des bâtiments est toujours très actifs. Plusieurs entreprises ont senti le potentiel économique du positionnement *indoor* et investissent dans ce domaine. Parmi toutes les technologies, le WiFi est le plus utilisé en combinaison avec des algorithmes de fingerprinting. Aussi, de plus en plus de méthodes géométriques sont utilisées car certains constructeurs offrent des cartes et des routeurs WiFi dont les API (*Application Programming Interface*) permettent de déterminer le temps d'arrivée ou l'angle d'arrivée. Après le WiFi vient le Bluetooth, puis le RFID/NFC. Cependant, ces systèmes commerciaux ciblent en majorité les personnes non handicapées avec des coûts d'installation assez élevés.

## Perspectives

Le diagramme, illustré par la figure 6.5, montre la démarche expérimentale que nous avons retenue pour tester la méthode API. En partant des règles de fonctionnement des fourmis *P. apicalis*, nous avons mis au point une méthode de clustering semi-supervisée. Nous avons utilisé un certain nombre de bases de données, d'indices d'évaluation et divers valeurs des paramètres pour évaluer les résultats fournis par API. En ce qui concerne l'application des fourmis au problème de classification nous pouvons envisager deux cas de figure : transformer le problème en problème de clustering et appliquer API ou à partir du paradigme des fourmis *P. apicalis* déterminer une modélisation du problème de clustering. D'un autre côté, nous avons étudié le positionnement dans les bâtiments ainsi que la façon de l'appliquer aux personnes handicapées.

En perspectives, plusieurs points nécessitent d'être améliorés :

- continuer le développement et les recherches autour de la version non supervisée de API. Nous avons donné plusieurs pistes, aussi bien pour le recrutement que la création des nids. Il sera intéressant d'étudier d'autres modèles de recrutement et de création des nids ;
- étudier de façon analytique les fonctions de densité basées sur le paramètre  $\alpha$ . Les résultats expérimentaux montrent qu'une valeur de  $\alpha = 0.5$  donne de meilleurs résultats dans la plupart des cas. Il faudra déterminer la valeur optimale à utiliser pour une base de données particulière ;
- implémenter plus d'indices d'évaluations et aussi exploiter les idées provenant des autres paradigmes de classification ; dans ce document, nous avons approfondi plusieurs de ces idées comme les indices de liens utilisés par les méthodes hiérarchiques ;
- appliquer API au positionnement interne ; nous avons présenté quelques pistes qui sont à approfondir tant les algorithmes à intelligence collective sont prometteurs pour les réseaux distribués et l'informatique ubiquitaire ;
- en ce qui concerne le positionnement interne des personnes handicapées :
  - continuer le développement de notre système de positionnement décrit dans la première proposition ; En effet, il sera intéressant d'étudier plusieurs modèles de propagation, la calibration de ces modèles par les valeurs déterminées dans l'environnement. La simulation du fingerprinting permettra aussi de comprendre plusieurs aspects de la localisation interne ;
  - continuer le développement du middleware et aussi la simulation des cas que

## CONCLUSION

---

- nous avons présentés en utilisant notamment les éléments finis ;
- le déploiement des systèmes de positionnement dans des situations réelles en collaboration avec des personnels de santé, des maisons de retraites, etc. ;

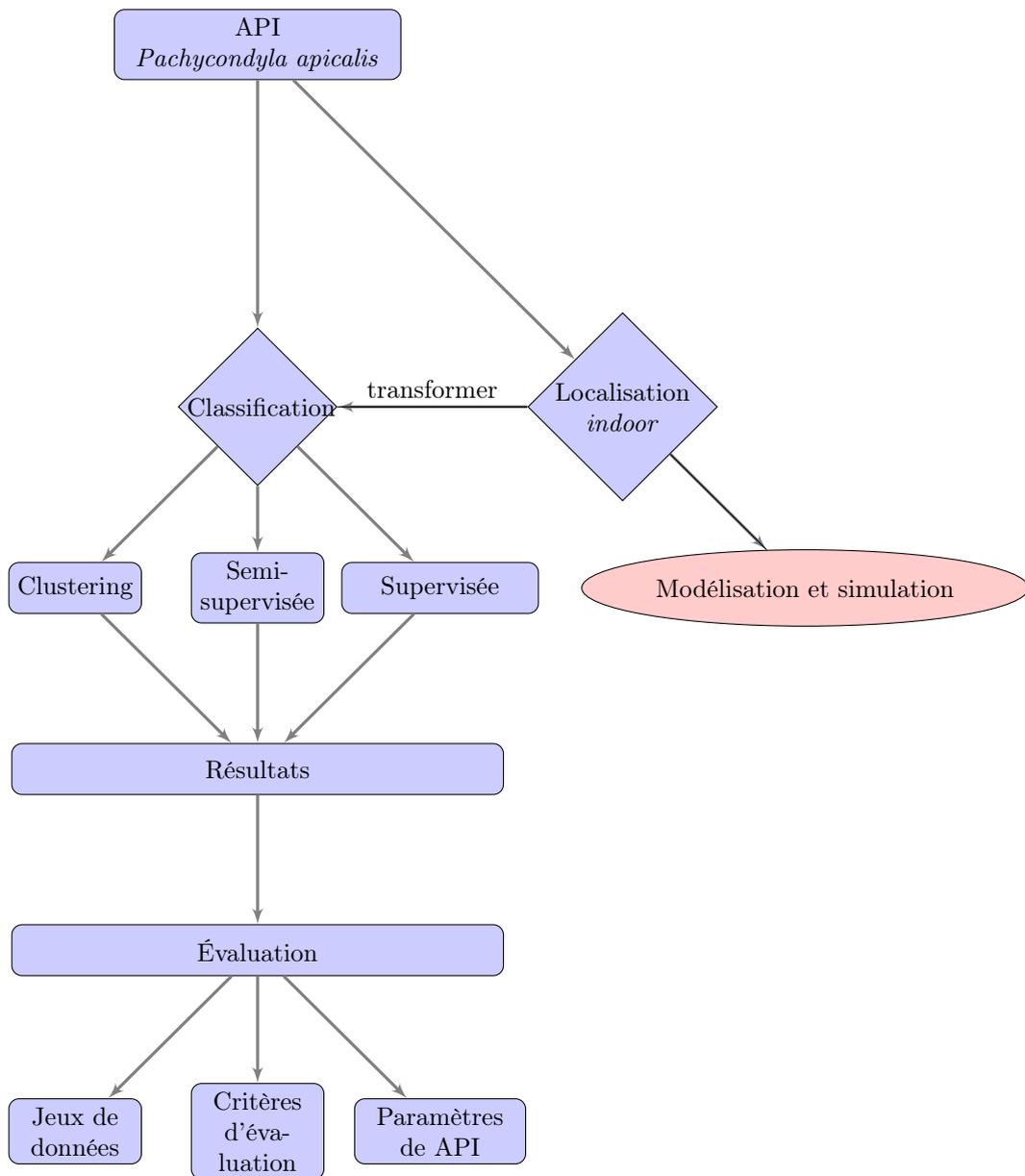


FIGURE 6.5 – Diagramme des expérimentations pour la méthode API.

## CONCLUSION

---

# Annexes



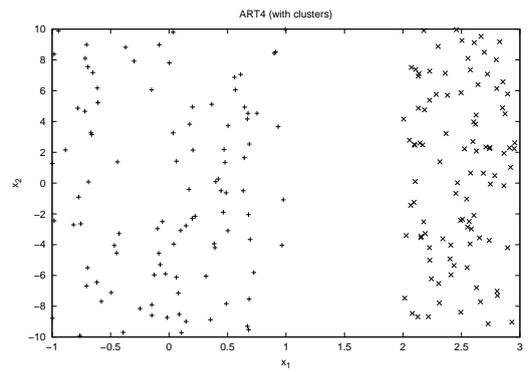
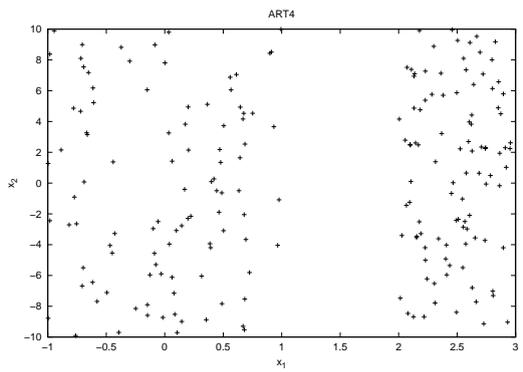
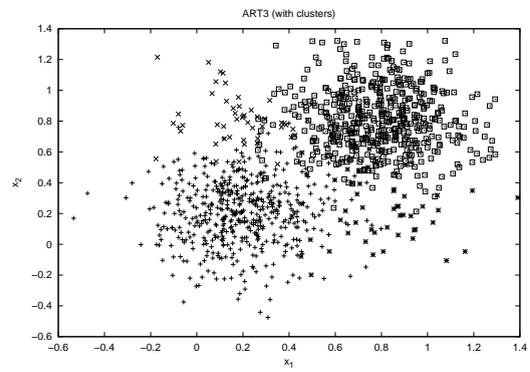
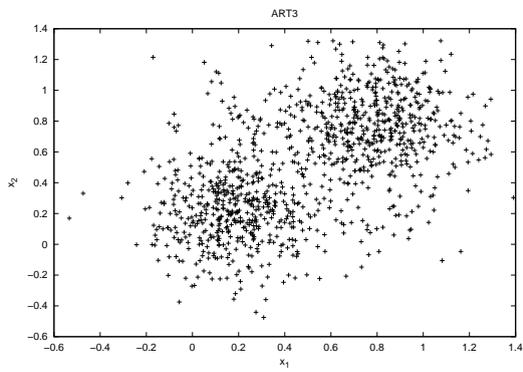
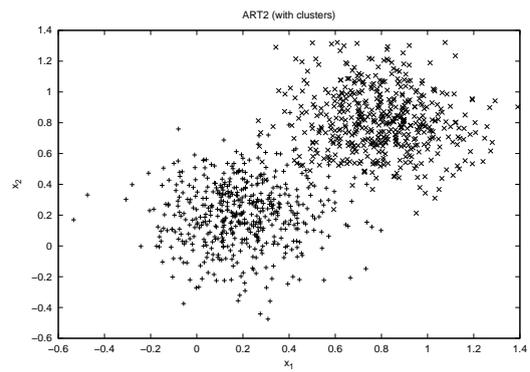
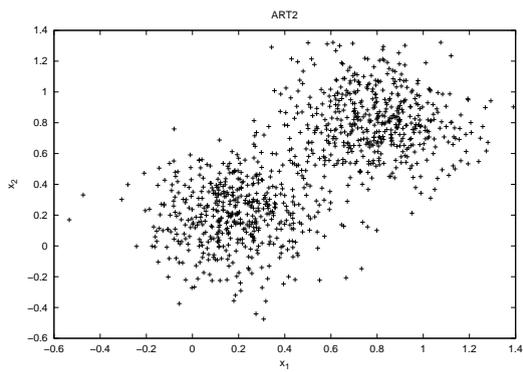
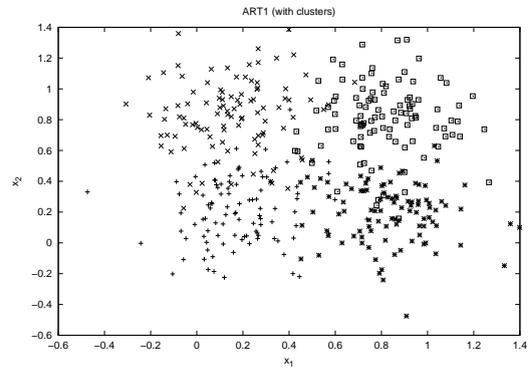
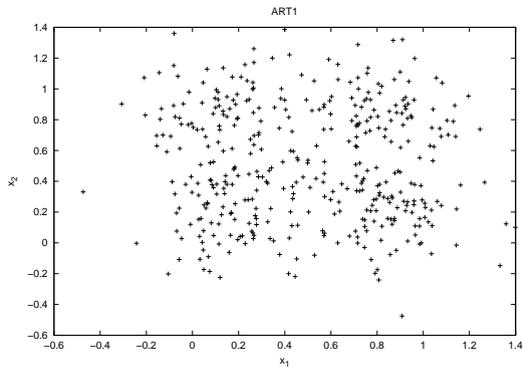
# Annexe A

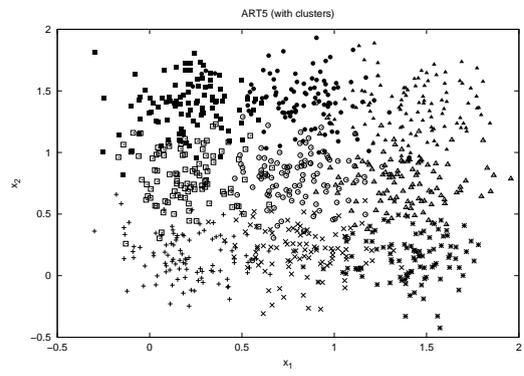
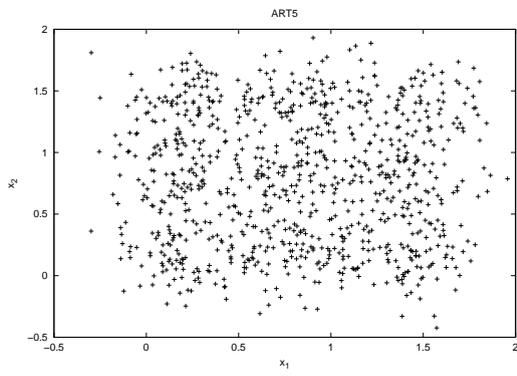
## Description des données aléatoires

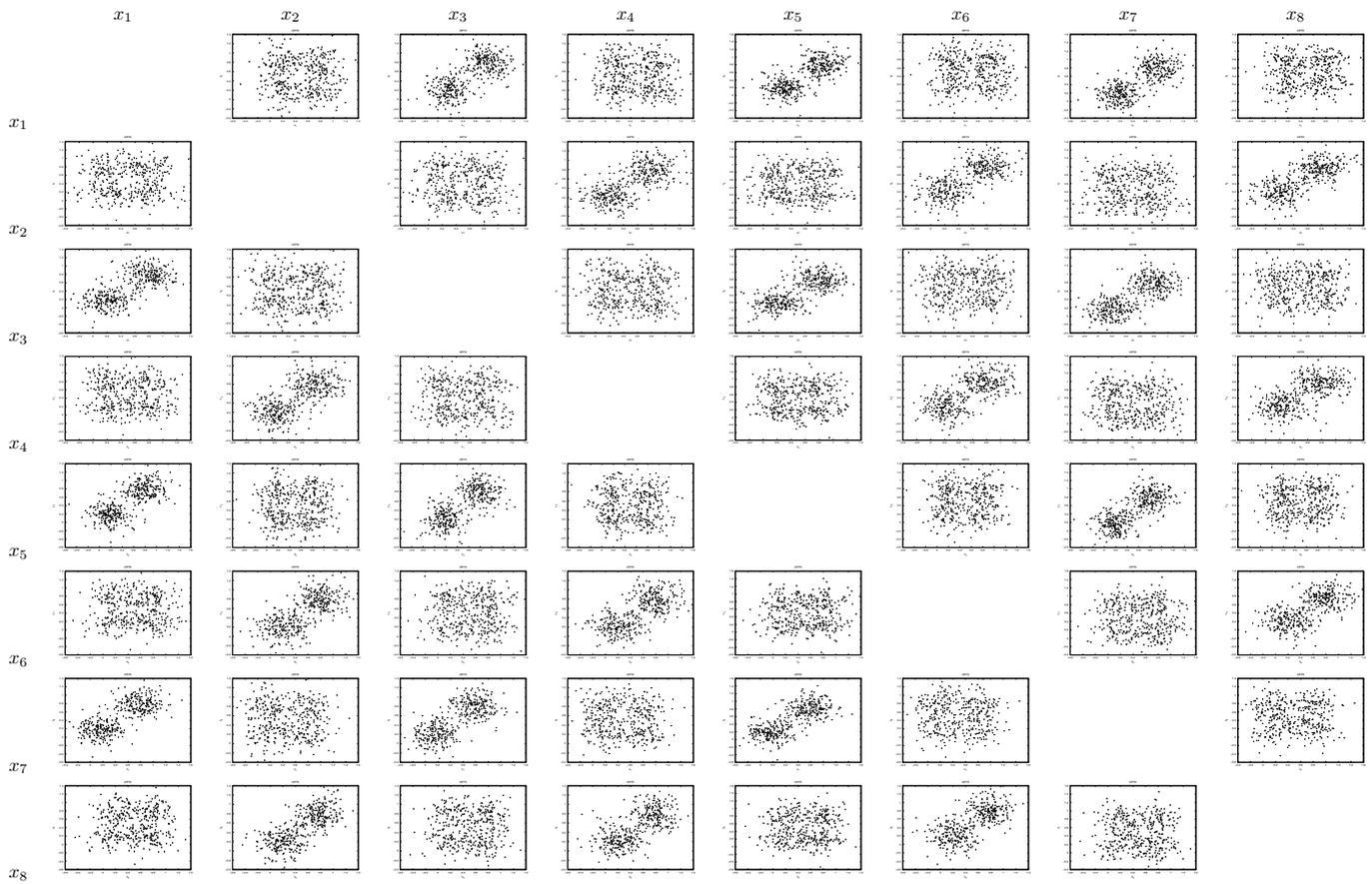
Dans cette partie nous allons illustrer les données aléatoires utilisées dans le chapitre 6. Nous allons utiliser, ces bases sont rassemblées dans la table A.1.

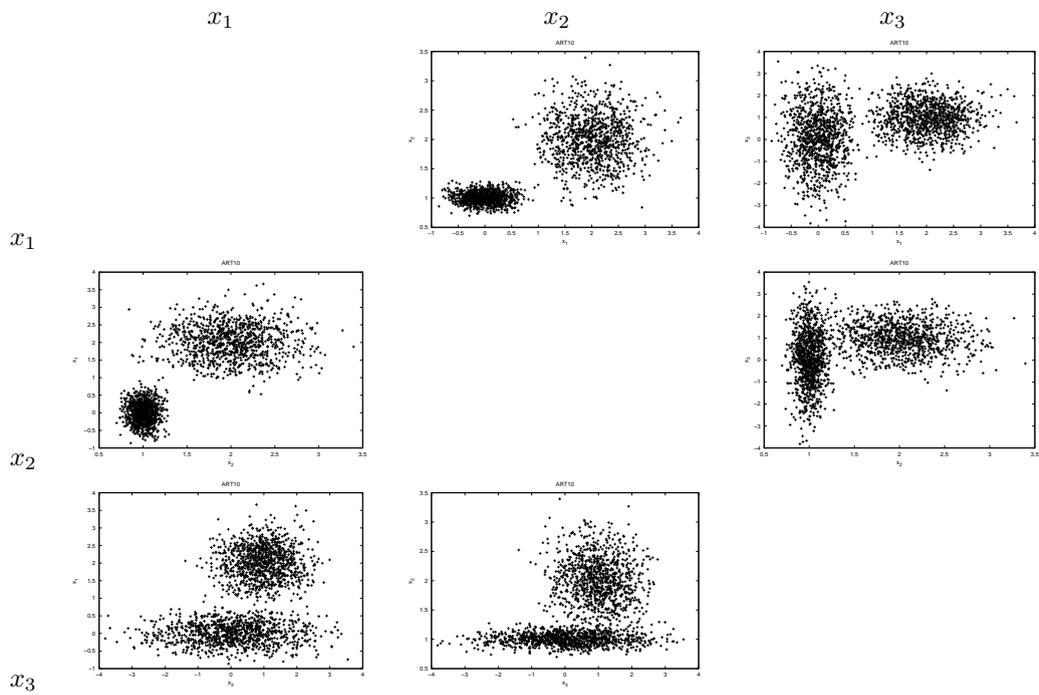
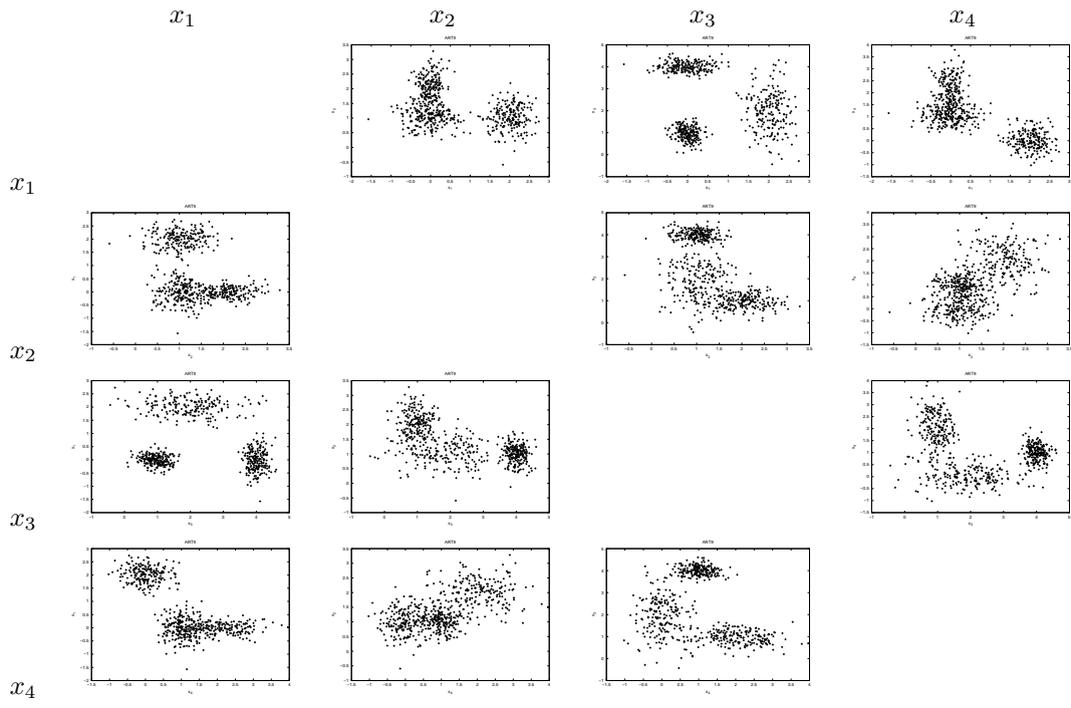
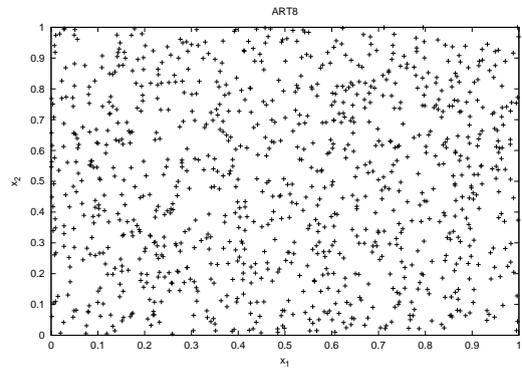
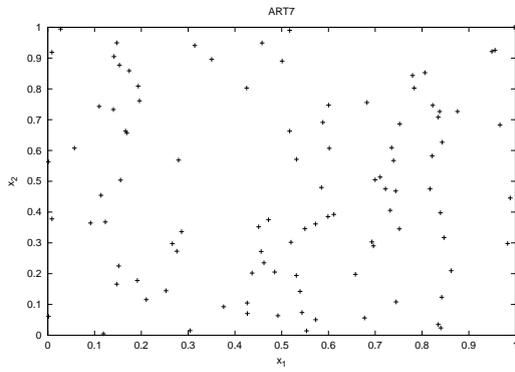
Nom	M	K	$ c_i $	répartition des points
art1	2	4	100	$(\mathcal{N}(0.2, 0.2), \mathcal{N}(0.2, 0.2))$ $(\mathcal{N}(0.2, 0.2), \mathcal{N}(0.8, 0.2))$ $(\mathcal{N}(0.8, 0.2), \mathcal{N}(0.2, 0.2))$ $(\mathcal{N}(0.8, 0.2), \mathcal{N}(0.8, 0.2))$
art2	2	2	500	$(\mathcal{N}(0.2, 0.2), \mathcal{N}(0.2, 0.2))$ $(\mathcal{N}(0.8, 0.2), \mathcal{N}(0.8, 0.2))$
art3	2	4	500	$(\mathcal{N}(0.2, 0.2), \mathcal{N}(0.2, 0.2))$ $(\mathcal{N}(0.2, 0.2), \mathcal{N}(0.8, 0.2))$ 50 $(\mathcal{N}(0.8, 0.2), \mathcal{N}(0.2, 0.2))$ 500 $(\mathcal{N}(0.8, 0.2), \mathcal{N}(0.8, 0.2))$
art4	2	2	100	$(\mathcal{U}[-1, 1], \mathcal{U}[-10, 10])$ 100 $(\mathcal{U}[2, 3], \mathcal{U}[-10, 10])$
art5	2	9	100	$(\mathcal{N}(0.2, 0.2), \mathcal{N}(0.2, 0.2))$ $(\mathcal{N}(0.8, 0.2), \mathcal{N}(0.2, 0.2))$ $(\mathcal{N}(1.4, 0.2), \mathcal{N}(0.2, 0.2))$  $\vdots$  100 $(\mathcal{N}(1.4, 0.2), \mathcal{N}(1.4, 0.2))$
art6	8	4	100	$(\mathcal{N}(0.2, 0.2), \dots, \mathcal{N}(0.2, 0.2))$ $(\mathcal{N}(0.2, 0.2), \mathcal{N}(0.8, 0.2), \dots, \mathcal{N}(0.2, 0.2), \mathcal{N}(0.8, 0.2))$ $(\mathcal{N}(0.8, 0.2), \mathcal{N}(0.2, 0.2), \dots, \mathcal{N}(0.8, 0.2), \mathcal{N}(0.2, 0.2))$ $(\mathcal{N}(0.8, 0.2), \dots, \mathcal{N}(0.8, 0.2))$
art7	2	1		$(\mathcal{U}[0, 1], \mathcal{U}[0, 1])$
art8	2	1	1000	$(\mathcal{U}[0, 1], \mathcal{U}[0, 1])$
art9	4	3	200	$(\mathcal{N}(0.0, 0.2), \mathcal{N}(2.0, 0.5), \mathcal{N}(1.0, 0.3), \mathcal{N}(2.0, 0.7))$ $(\mathcal{N}(0.0, 0.4), \mathcal{N}(1.0, 0.3), \mathcal{N}(4.0, 0.2), \mathcal{N}(1.0, 0.3))$ $(\mathcal{N}(2.0, 0.3), \mathcal{N}(1.0, 0.4), \mathcal{N}(2.0, 0.9), \mathcal{N}(0.0, 0.4))$
art10	3	2	1000	$(\mathcal{N}(0.0, 0.3), \mathcal{N}(1.0, 0.1), \mathcal{N}(0.0, 1.2))$ $(\mathcal{N}(2.0, 0.5), \mathcal{N}(2.0, 0.4), \mathcal{N}(1.0, 0.7))$

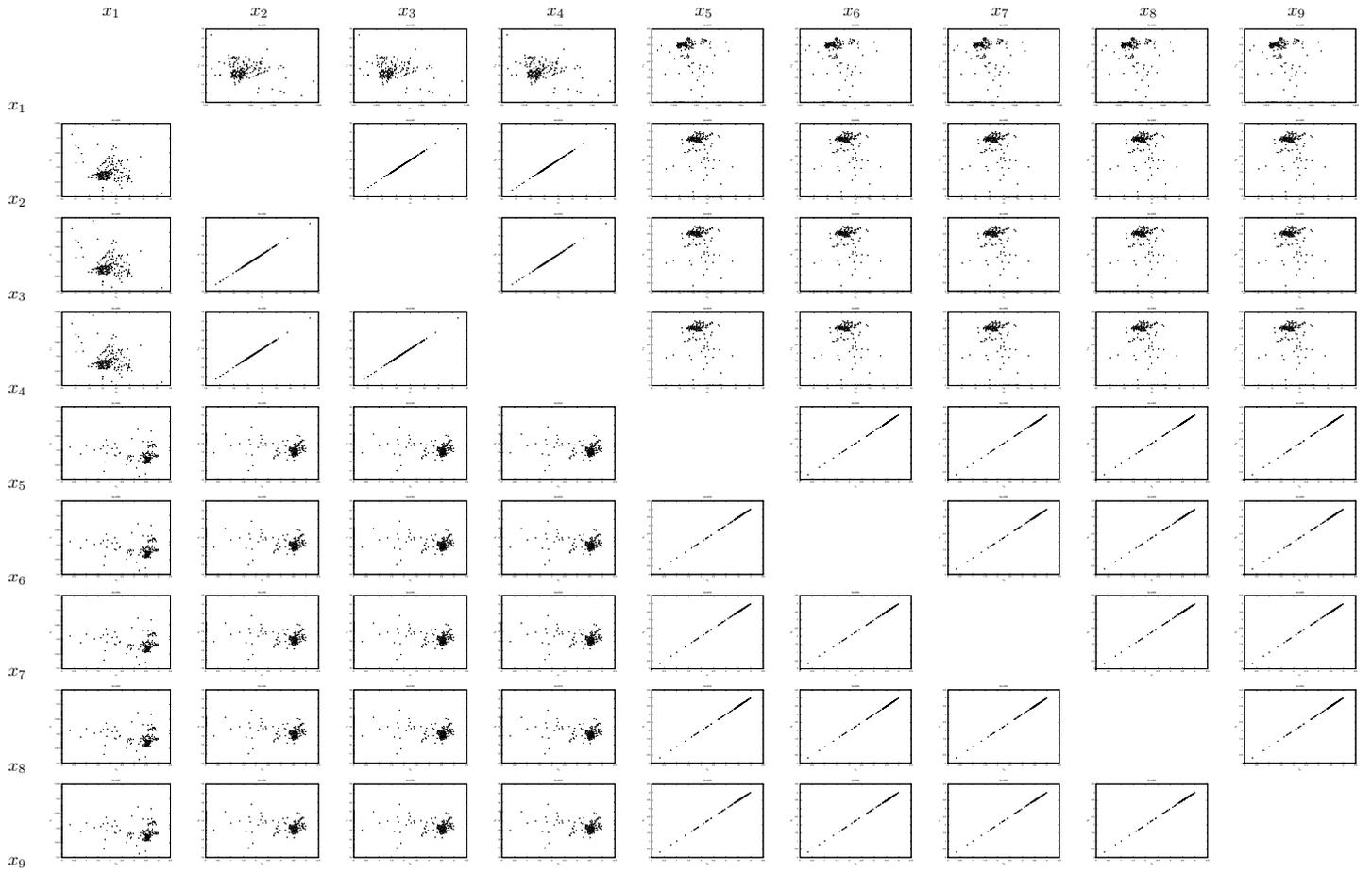
TABLE A.1 – Caractéristiques des données aléatoires. M est le nombre d'attributs, K le nombre de classes et  $|c_i|$  le nombre d'objets dans chaque classe [Monmarché *et al.*, 2001].

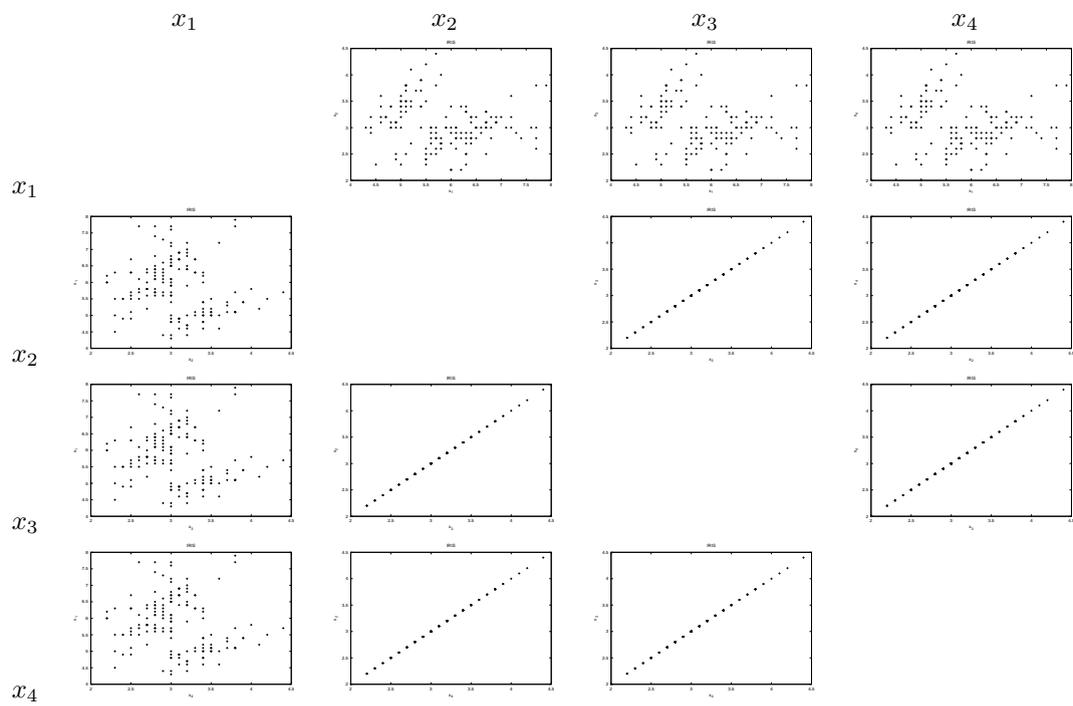














# Bibliographie

- [Abascal *et al.*, 2009] ABASCAL, J., BONAIL, B., GARDEAZABAL, L., LAFUENTE, A. et SALVADOR, Z. (2009). Managing intelligent services for people with disabilities and elderly people. In *Universal Access in Human-Computer Interaction. Intelligent and Ubiquitous Interaction Environments*, pages 623–630. Springer.
- [Abbasi et Younis, 2007] ABBASI, A. A. et YOUNIS, M. F. (2007). A survey on clustering algorithms for wireless sensor networks. *Computer Communications*, 30(14):2826–2841.
- [Abeywickrama et Ramakrishnan, 2012] ABEYWICKRAMA, D. et RAMAKRISHNAN, S. (2012). Context-aware services engineering : models, transformations, and verification. *ACM Transactions on Internet Technology (TOIT)*, 11(3):10.
- [Admane *et al.*, 2006] ADMANE, L., BENATCHBA, K., KOUDIL, M., SIAD, L. et MAZIZ, S. (2006). Antpart : an algorithm for the unsupervised classification problem using ants. *Applied Mathematics and Computation*, 180(1):16–28.
- [Agrawal, 2011] AGRAWAL, D. P. (2011). Designing wireless sensor networks : from theory to applications. *Central Europ. J. Computer Science*, 1(1):2–18.
- [Agrawal et Patwari, 2009] AGRAWAL, P. et PATWARI, N. (2009). Correlated link shadow fading in multi-hop wireless networks. *Wireless Communications, IEEE Transactions on*, 8(8):4024–4036.
- [Akyildiz *et al.*, 2002] AKYILDIZ, I. F., SU, W., SANKARASUBRAMANIAM, Y. et CAYIRCI, E. (2002). Wireless sensor networks : a survey. *Computer networks*, 38(4):393–422.
- [Amadou Kountché *et al.*, 2012] AMADOU KOUNTCHÉ, D., MONMARCHÉ, N. et SLIMANE, M. (2012). The pachycondyla apicalis ants search strategy for data clustering problems. *Swarm and Evolutionary Computation*, pages 3–11.
- [Amadou Kountché *et al.*, 2011] AMADOU KOUNTCHÉ, D., NEVEUX, B., MONMARCHÉ, N., GAUCHER, P. et SLIMANE, M. (2011). Indoor localization and guidance system for disabled people. In *Network of the Future (NOF), 2011 International Conference on the*, page 81–86. IEEE.
- [Apache Foundation, 2013] APPACHE FOUNDATION (2013). Apache shindig. <http://shindig.apache.org/>. [En ligne ; Consulté le 16 Août 2013].
- [Asch, 2010] ASCH, G. (2010). *Les capteurs en instrumentation industrielle*. Dunod, 7-ème édition.
- [Aschenbruck *et al.*, 2010] ASCHENBRUCK, N., ERNST, R., GERHARDS-PADILLA, E. et SCHWAMBORN, M. (2010). Bonnmotion : a mobility scenario generation and analysis

- tool. In *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*, page 51. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
- [Atzori *et al.*, 2010] ATZORI, L., IERA, A. et MORABITO, G. (2010). The internet of things : A survey. *Computer Networks*, 54(15):2787–2805.
- [Bahl et Padmanabhan, 2000] BAHL, P. et PADMANABHAN, V. (2000). RADAR : An in-building RF-based user location and tracking system. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 775–784. IEEE.
- [Ball et Hall, 1965] BALL, G. et HALL, D. (1965). ISODATA, a novel method of data analysis and classification. Rapport technique, Stanford University.
- [Baronti *et al.*, 2007] BARONTI, P., PILLAI, P., CHOOK, V. W., CHessa, S., GOTTA, A. et HU, Y. F. (2007). Wireless sensor networks : A survey on the state of the art and the 802.15. 4 and zigbee standards. *Computer communications*, 30(7):1655–1695.
- [Bejuri *et al.*, 2011] BEJURI, W. M. Y. W., MOHAMAD, M. M. et SAPRI, M. (2011). Ubiquitous Positioning : A Taxonomy for Location Determination on Mobile Navigation System. *arXiv preprint arXiv :1103.5035*, page 15.
- [Bekkelien *et al.*, 2012] BEKKELIEN, A., DERIAZ, M. et MARCHAND-MAILLET, S. (2012). Bluetooth indoor positioning. Mémoire de D.E.A., Université de Genève.
- [Benedicto *et al.*, 2000] BENEDICTO, J., DINWIDDY, S., GATTI, G., LUCAS, R. et LUGERT, M. (2000). Galileo : Satellite system design. *European Space Agency*.
- [Bensky, 2007] BENSKY, A. (2007). *Wireless positioning technologies and applications*. Artech House, Inc.
- [Beresford et Stajano, 2004] BERESFORD, A. et STAJANO, F. (2004). Mix zones : User privacy in location-aware services. In *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, pages 127–131. IEEE.
- [Bezdek, 1981] BEZDEK, J. C. (1981). Models for pattern recognition. In *Pattern Recognition with Fuzzy Objective Function Algorithms*, pages 1–13. Springer.
- [Bluetooth, SIG, 2013] BLUETOOTH, SIG (2013). Bluetooth core version 4.0. <https://www.bluetooth.org/en-us/specification/adopted-specifications/>. [En ligne; Consulté le 16 Août 2013].
- [Bolliger, 2008] BOLLIGER, P. (2008). Redpin-adaptive, zero-configuration indoor localization through user collaboration. In *Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments*, pages 55–60. ACM.
- [Bonabeau *et al.*, 1999] BONABEAU, E., DORIGO, M. et THERAULAZ, G. (1999). *From Natural to Artificial Swarm Intelligence*. Oxford University Press.
- [BonnMotion Team, 2013] BONNMOTION TEAM (2013). Bonnmotion. <http://net.cs.uni-bonn.de/wg/cs/applications/bonnmotion/>. [En ligne; Consulté le 16 Août 2013].

- [Boorman et Arabie, 1972] BOORMAN, S. A. et ARABIE, P. (1972). Structural Measures and the Method of Sorting. In SHEPARD, R. N., ROMNEY, A. K. et NERLOVE, S. B., éditeurs : *Multidimensional Scaling Theory and Applications in the Behavioral Sciences 1 Theory*, pages 225–249. Seminar Press.
- [Boryczka, 2008] BORYCZKA, U. (2008). Ant clustering algorithm. *Intelligent Information Systems*, 1998:455–458.
- [Boryczka, 2009] BORYCZKA, U. (2009). Finding groups in data : Cluster analysis with ants. *Applied Soft Computing*, 9(1):61–70.
- [Brucker et Barthélemy, 2007] BRUCKER, F. et BARTHÉLEMY, J. (2007). *Eléments de classification : aspects combinatoires et algorithmiques*. Collection Méthodes stochastiques appliquées. Hermès Science Publications.
- [Brunato et Battiti, 2005] BRUNATO, M. et BATTITI, R. (2005). Statistical learning theory for location fingerprinting in wireless lans. *Computer Networks*, 47(6):825–845.
- [Bulusu et al., 2000] BULUSU, N., HEIDEMANN, J. et ESTRIN, D. (2000). Gps-less low-cost outdoor localization for very small devices. *Personal Communications, IEEE*, 7(5):28–34.
- [C. J. van Rijsbergen, 1979] C. J. VAN RIJSBERGEN (1979). Information retrieval. <http://www.dcs.gla.ac.uk/Keith/Preface.html>. [En ligne ; Consulté le 19 septembre 2013].
- [Cao et al., 2006] CAO, F., ESTER, M., QIAN, W. et ZHOU, A. (2006). Density-based clustering over an evolving data stream with noise. In *SDM*.
- [Carlson, 2013] CARLSON, D. (2013). Ambien dynamix. <http://ambientdynamix.org>. [En ligne ; Consulté le 16 Août 2013].
- [Chan et Ho, 1994] CHAN, Y. et HO, K. (1994). A simple and efficient estimator for hyperbolic location. *Signal Processing, IEEE Transactions on*, 42(8):1905–1915.
- [Chen et al., 2003] CHEN, M., CHENG, F. et GUDAVALLI, R. (2003). Precision and accuracy in an indoor localization system. Rapport technique, University of California, Berkeley.
- [Cheung et al., 2006] CHEUNG, K. C., INTILLE, S. S. et LARSON, K. (2006). An inexpensive bluetooth-based indoor positioning hack. *Proc. UbiComp06 Extended Abstracts*.
- [Chinchor et Sundheim, 1993] CHINCHOR, N. et SUNDHEIM, B. (1993). Muc-5 evaluation metrics. In *Proceedings of the 5th conference on Message understanding*, pages 69–78. Association for Computational Linguistics.
- [Collette et Siarry, 2003] COLLETTE, Y. et SIARRY, P. (2003). *Multiobjective optimization : principles and case studies*. Springer.
- [Cornuéjols et Miclet, 2010] CORNUÉJOLS, A. et MICLET, L. (2010). *Apprentissage artificiel : Concepts et algorithmes*. Eyrolles.
- [Corral et al., 2008] CORRAL, P., PEÑA, E., GARCIA, R., ALMENAR, V. et de C LIMA, A. (2008). Distance estimation system based on zigbee. In *Computational Science and Engineering Workshops, 2008. CSEWORKSHOPS'08. 11th IEEE International Conference on*, pages 405–411. IEEE.

- [Dakkak *et al.*, 2011] DAKKAK, M., NAKIB, A., DAACHI, B., SIARRY, P. et LEMOINE, J. (2011). Indoor localization method based on rtt and aoa using coordinates clustering. *Computer Networks*, 55(8):1794 – 1803.
- [Dargie et Poellabauer, 2011] DARGIE, W. et POELLABAUER, C. (2011). Fundamentals of wireless sensor networks : Theory and practice. *Fundamentals of Wireless Sensor Networks*.
- [Darnell et Douglas, 1991] DARNELL, R. et DOUGLAS, C. (1991). Cellular position locating system. US Patent 5,043,736.
- [Deneubourg *et al.*, 1991] DENEUBOURG, J.-L., GOSS, S., FRANKS, N., SENDOVA-FRANKS, A., DETRAIN, C. et CHRÉTIEN, L. (1991). The dynamics of collective sorting robot-like ants and ant-like robots. In *Proceedings of the first international conference on simulation of adaptive behavior on From animals to animats*, pages 356–363.
- [Dennis et Moré, 1977] DENNIS, Jr, J. E. et MORÉ, J. J. (1977). Quasi-newton methods, motivation and theory. *SIAM review*, 19(1):46–89.
- [Dhillon *et al.*, 2004] DHILLON, I. S., GUAN, Y. et KULIS, B. (2004). Kernel k-means : spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 551–556, New York, NY, USA. ACM.
- [Digi Inc, 2013] DIGI INC (2013). Description des modules xbee. <http://www.digi.com/xbee/>. [En ligne ; Consulté le 16 Août 2013].
- [DILA, 2013] DILA (2013). Vie public. <http://www.vie-publique.fr>. Edité par la Direction de l'information légale et administrative [En ligne ; Consulté le 16 Août 2013].
- [Doherty *et al.*, 2001] DOHERTY, L., EL GHAOU, L. *et al.* (2001). Convex position estimation in wireless sensor networks. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1655–1663. IEEE.
- [Druppel Fondation, 2013] DRUPPAL FONDATION (2013). Druppel CMS. <http://www.druppel.org>. [En ligne ; Consulté le 16 Août 2013].
- [Dunn, 1973] DUNN, J. C. (1973). A fuzzy relative of the ISODATA process and its use in detecting. *Journal Of Cybernetics*, 3:32–57.
- [Dunn, 1974] DUNN, J. C. (1974). Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1):95–104.
- [Ekahau, 2013] EKAHAU (2013). Ekahau Real-Time Location System. <http://www.ekahau.com>. [En ligne ; Consulté le 16 Août 2013].
- [ERCIM, 2011] ERCIM (2011). ERCIM News. Special theme : Ambient Assisted Living. <http://ercim-news.ercim.eu/en87>. [En ligne ; Consulté le 16 Août 2013].
- [Ester *et al.*, 1996] ESTER, M., XU, X., KRIEGEL, H.-p. et SANDER, J. (1996). Density-based algorithm for discovering clusters in large spatial databases with noise. In SIMOUDIS, E., HAN, J. et FAYYAD, U., éditeurs : *Proceedings Of The Acm Sigkdd International Conference On Knowledge Discovery And Data Mining*, volume pages, pages 226–231. AAAI.

- [Evennou, 2007] EVENNOU, F. (2007). *Techniques et technologies de localisation avancées pour terminaux mobiles dans les environnements indoor*. Thèse de doctorat, Université Joseph-Fourier-Grenoble I.
- [F. et Rohlf, 1982] F. et ROHLF, J. (1982). Consensus indices for comparing classifications. *Mathematical Biosciences*, 59(1):131 – 144.
- [Fall et Varadhan, 2002] FALL, K. et VARADHAN, K. (2002). *The ns Manual (formerly ns Notes and Documentation)*.
- [Feldmann et al., 2003] FELDMANN, S., KYAMAKYA, K., ZAPATER, A. et LUE, Z. (2003). An indoor bluetooth-based positioning system : concept, implementation and experimental evaluation. *In International Conference on Wireless Networks*, pages 109–113.
- [Ferguson, 1973] FERGUSON, T. S. (1973). A Bayesian Analysis of Some Nonparametric Problems. *Annals of Statistics*, 1(2):209–230.
- [Figueiredo et Jain, 2002] FIGUEIREDO, M. A. F. et JAIN, A. K. (2002). Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):381–396.
- [Fleury et Simplot-Ryl, 2009] FLEURY, E. et SIMPLOT-RYL, D. (2009). *Réseaux de capteurs : théorie et modélisation*. Architecture, Applications, Service. Hermes.
- [Fonlupt et al., 1993] FONLUPT, C., PREUX, P. et ROBILLIARD, D. (1993). Preventing premature convergence via cooperating genetic algorithms. *In Proc. of the Workshop on Foundations of Genetic Algorithms*. Citeseer.
- [Forgy, 1965] FORGY, E. (1965). Cluster analysis of multivariate data : Cluster analysis of multivariate data : efficiency vs interpretability of classifications. *Biometrics*, 21:768–769.
- [Fox et al., 2003] FOX, V., HIGHTOWER, J., LIAO, L., SCHULZ, D. et BORRIELLO, G. (2003). Bayesian filtering for location estimation. *Pervasive Computing, IEEE*, 2(3):24–33.
- [Fraunhofer IIS, 2013] FRAUNHOFER IIS (2013). Fraunhofer-Institut für Integrierte Schaltungen IIS. <http://www.iis.fraunhofer.de>. [En ligne ; Consulté le 16 Août 2013].
- [Garcia et al., 2009] GARCIA, M., MONTIEL, O., CASTILLO, O., SEPÚLVEDA, R. et MELIN, P. (2009). Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation. *Applied Soft Computing*, 9(3):1102–1110.
- [Genco, 2004] GENCO, A. (2004). Bluetooth positioning optimization by genetic algorithm. *WSEAS Transactions on Information Science and Applications*, 1(6):1584–1590.
- [Gentile et al., 2012] GENTILE, C., ALSINDI, N., RAUFELS, R. et TEOLIS, C. (2012). *Geolocation techniques : principles and applications (hardback)*. Springer.
- [Gezici et al., 2005] GEZICI, S., TIAN, Z., GIANNAKIS, G., KOBAYASHI, H., MOLISCH, A., POOR, H. et SAHINOGLU, Z. (2005). Localization via ultra-wideband radios : a look at positioning aspects for future sensor networks. *Signal Processing Magazine, IEEE*, 22(4):70–84.

- [Gómez *et al.*, 2010] GÓMEZ, C., PARADELLS, J. et CABALLERO, J. E. (2010). *Sensors Everywhere Wireless Network Technologies and Solutions*. Fondation Vodafone Espane.
- [Google Inc, 2011] GOOGLE INC (2011). Go indoors With Google Maps 6.0. <http://googlemobile.blogspot.fr/2011/11/go-indoors-with-google-maps-60-for.html>. [En ligne; Consulté le 16 Août 2013].
- [Grewal *et al.*, 2007] GREWAL, M., WEILL, L. et ANDREWS, A. (2007). *Global positioning systems, inertial navigation, and integration*. Wiley-Interscience.
- [Gupta et Hirdesh, 2007] GUPTA, S. et HIRDESH, A. (2007). Overview of m2m. [http://sites.google.com/site/hridayankit/M2M\\_overview\\_paper.pdf](http://sites.google.com/site/hridayankit/M2M_overview_paper.pdf). [En ligne; Consulté le 16 Août 2013].
- [Gustafsson et Gunnarsson, 2005] GUSTAFSSON, F. et GUNNARSSON, F. (2005). Mobile Positioning Using Wireless Networks. *IEEE Signal Processing Mag*, 22:41–53.
- [Han *et al.*, 2006] HAN, J., KAMBER, M. et PEI, J. (2006). *Data Mining : Concepts and Techniques, Second Edition*. Morgan Kaufmann.
- [Handl *et al.*, 2003] HANDL, J., KNOWLES, J. D. et DORIGO, M. (2003). On the performance of ant-based clustering. *In HIS*, pages 204–213.
- [Handl et Meyer, 2002] HANDL, J. et MEYER, B. (2002). Improved ant-based clustering and sorting in a document retrieval interface. *In Parallel Problem Solving from Nature—PPSN VII*, pages 913–923. Springer.
- [Handl et Meyer, 2007] HANDL, J. et MEYER, B. (2007). Ant-based and swarm-based clustering. *Swarm Intelligence*, 1(2):95–113.
- [Hansen et Yu, 2001] HANSEN, M. H. et YU, B. (2001). Model Selection and the Principle of Minimum Description Length. *Journal of the American Statistical Association*, 96(454):746–774.
- [Hashemi, 1993] HASHEMI, H. (1993). The indoor radio propagation channel. *Proceedings of the IEEE*, 81(7):943–968.
- [He *et al.*, 2003] HE, T., HUANG, C., BLUM, B., STANKOVIC, J. et ABDELZAHER, T. (2003). Range-free localization schemes for large scale sensor networks. *In Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 81–95. ACM.
- [Hightower et Borriello, 2001] HIGHTOWER, J. et BORRIELLO, G. (2001). Location systems for ubiquitous computing. *Computer*, 34(8):57–66.
- [Hinneburg et Gabriel, 2007] HINNEBURG, A. et GABRIEL, H.-H. (2007). Denclue 2.0 : fast clustering based on kernel density estimation. *In Proceedings of the 7th international conference on Intelligent data analysis, IDA'07*, pages 70–80, Berlin, Heidelberg. Springer-Verlag.
- [Hofmann-Wellenhof *et al.*, 2007] HOFMANN-WELLENHOF, B., LICHTENEGGER, H. et WASLE, E. (2007). *GNSS—Global Navigation Satellite Systems : GPS, GLONASS, Galileo & more*. Springer.
- [Hore *et al.*, 2009] HORE, P., HALL, L. O., GOLDFOF, D. B., GU, Y., MAUDSLEY, A. A. et DARKAZANLI, A. (2009). A scalable framework for segmenting magnetic resonance images. *J. Signal Process. Syst.*, 54(1-3):183–203.

- [Hornung *et al.*, 2010] HORNUNG, A., WURM, K. M. et BENNEWITZ, M. (2010). Humanoid robot localization in complex indoor environments. *In Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1690–1695. IEEE.
- [Hubert et Arabie, 1985] HUBERT, L. et ARABIE, P. (1985). Comparing partitions. *Journal of Classification*, 2(1):193–218.
- [IEEE Computer Society, 2005] IEEE COMPUTER SOCIETY (2005). Ieee 802.15.1. Rapport technique June, IEEE.
- [Jain, 2010] JAIN, A. K. (2010). Data clustering : 50 years beyond K-means. *Pattern Recognition Letters*, 31(8):651–666.
- [Jain *et al.*, 1999] JAIN, a. K., MURTY, M. N. et FLYNN, P. J. (1999). Data clustering : a review. *ACM Computing Surveys*, 31:264–323.
- [Jain *et al.*, 2004] JAIN, A. K., TOPCHY, A., LAW, M. H. C. et BUHMANN, J. M. (2004). Landscape of clustering algorithms. *In Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 1 - Volume 01*, ICPR '04, pages 260–263, Washington, DC, USA. IEEE Computer Society.
- [JCGM, 2008] JCGM (2008). *International Vocabulary of Metrology : Basic and General Concepts and Associated Terms (VIM)*. Working Group 2 of the Joint Committee for Guides in Metrology (JCGM/WG 2), BIPM, Pavillon de Breteuil F-92312 Sèvres Cedex, FRANCE, 3rd edition édition.
- [Jiang et Leung, 2007] JIANG, Y. et LEUNG, V. C. (2007). An asymmetric double sided two-way ranging for crystal offset. *In Signals, Systems and Electronics, 2007. ISSSE'07. International Symposium on*, pages 525–528. IEEE.
- [Kangas *et al.*, 2011] KANGAS, A., SIOMINA, I. et WIGREN, T. (2011). Positioning in LTE. *Handbook of Position Location*, pages 1081–1127.
- [Kaplan et Hegarty, 2006] KAPLAN, E. et HEGARTY, C. (2006). *Understanding GPS : principles and applications*. Artech House Publishers.
- [Karl et Willig, 2007] KARL, H. et WILLIG, A. (2007). *Protocols and architectures for wireless sensor networks*. Wiley-Interscience.
- [Kashima *et al.*, 2003] KASHIMA, H., TSUDA, K. et INOKUCHI, A. (2003). Marginalized kernels between labeled graphs. *In ICML*, pages 321–328.
- [Khaled et Gijs, 2013] KHALED, N. et GIJS, M. (2013). Advanced Wireless Body Sensor Networks (WBSN) for Performance Monitoring in Sports. <http://si.epfl.ch/page-34870-en.html>. [En ligne; Consulté le 16 Août 2013].
- [Kinney *et al.*, 2003] KINNEY, P. *et al.* (2003). Zigbee technology : Wireless control that simply works. *In Communications design conference*, volume 2.
- [Kleinberger *et al.*, 2007] KLEINBERGER, T., BECKER, M., RAS, E., HOLZINGER, A. et MÜLLER, P. (2007). Ambient intelligence in assisted living : enable elderly people to handle future interfaces. *Universal access in human-computer interaction. Ambient interaction*, pages 103–112.
- [Kulkarni et Kumar, 2011] KULKARNI, R. et KUMAR, G. (2011). Technical correspondence. *IEEE Transactions on System, Man, and Cybernetics—Part C : Applications and Reviews*, 41(2):263.

- [Ledlie *et al.*, 2012] LEDLIE, J., PARK, J., CURTIS, D., CAVALCANTE, A., CAMARA, L., COSTA, A. et VIEIRA, R. (2012). Molé : A scalable, user-generated wifi positioning engine. *Journal of Location Based Services*, 6(2):55–80.
- [Lee, 2008] LEE, E. A. (2008). Cyber physical systems : Design challenges. In *Object Oriented Real-Time Distributed Computing (ISORC), 2008 11th IEEE International Symposium on*, pages 363–369. IEEE.
- [Lewis, 2004] LEWIS, F. (2004). Wireless sensor networks. *Smart Environments : Technologies, Protocols, and Applications*, pages 11–46.
- [Libelium, 2012] LIBELIUM (2012). Wasmote v1.1. <http://www.libelium.com/development-v11/>. [En ligne ; Consulté le 16 Août 2013].
- [Liu *et al.*, 2007] LIU, H., DARABI, H., BANERJEE, P. et LIU, J. (2007). Survey of wireless indoor positioning techniques and systems. *Systems, Man, and Cybernetics, Part C : Applications and Reviews, IEEE Transactions on*, 37(6):1067–1080.
- [Liu *et al.*, 2010] LIU, Y., YANG, Z., WANG, X. et JIAN, L. (2010). Location, localization, and localizability. *Journal of Computer Science and Technology*, 25(2):274–297.
- [Lohier et Présent, 2010] LOHIER, S. et PRÉSENT, D. (2010). *Transmissions et réseaux-5ème édition-Cours et exercices corrigés : Cours et exercices corrigés*. Dunod.
- [Lumer et Faieta, 1994] LUMER, E. et FAIETA, B. (1994). Diversity and adaptation in populations of clustering ants. In MEYER, J. A. et WILSON, S. W., éditeurs : *Proceedings of the Third International Conference on Simulation of Adaptive Behavior From Animals to Animats 3*, pages 501–508.
- [Mabrouk *et al.*, 2005] MABROUK, M., BYCHOWSKI, T., NIEDZWIADK, H., BISHR, Y., GAILLET, J., CRISP, N., WILBRINK, W., HORHAMMER, M., ROY, G. et MARGOULIS, S. (2005). OpenGIS location services (openLS) : Core services. *OGC Implementation Specification*, 5:016.
- [Macagnano *et al.*, 2012] MACAGNANO, D., DESTINO, G. et ABREU, G. (2012). A comprehensive tutorial on localization : Algorithms and performance analysis tools. *International Journal of Wireless Information Networks*, 19:1–25.
- [Madhavapeddy et Tse, 2005] MADHAVAPEDDY, A. et TSE, A. (2005). A study of bluetooth propagation using accurate indoor location mapping. *UbiComp 2005 : Ubiquitous Computing*, pages 903–903.
- [Madigan *et al.*, 2005] MADIGAN, D., EINAHRAWY, E., MARTIN, R. P., JU, W.-H., KRISHNAN, P. et KRISHNAKUMAR, A. (2005). Bayesian indoor positioning systems. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 2, pages 1217–1227. IEEE.
- [Malik, 2009] MALIK, A. (2009). *RTLS for Dummies*. Wiley.
- [Mao *et al.*, 2007] MAO, G., FIDAN, B. et ANDERSON, B. (2007). Wireless sensor network localization techniques. *Computer Networks*, 51(10):2529–2553.
- [Marco *et al.*, 2008] MARCO, A., CASAS, R., FALCO, J., GRACIA, H., ARTIGAS, J. et ROY, A. (2008). Location-based services for elderly and disabled people. *Computer Communications*, 31(6):1055–1066.

- [Martin *et al.*, 2013] MARTIN, J. A. G., RODRÍGUEZ, A. V. M., ZUBIETE, E. D., ROMERO, O. R. et GUILLÉN, S. M. (2013). Fingerprint indoor position system based. *Journal of Networks*, 8(1):37–44.
- [Mautz, 2009] MAUTZ, R. (2009). Overview of current indoor positioning systems. *Geodezija ir kartografija*, 35(1):18–22.
- [Mautz, 2012] MAUTZ, R. (2012). *Indoor Positioning Technologies*. Thèse de doctorat, Habil. ETH Zürich, 2012.
- [McCune, 2000] MCCUNE, E. (2000). Dsss vs. fhss narrowband interference performance issues. *RF Signal Processing Magazine*.
- [Mohanty et Parhi, 2013] MOHANTY, P. K. et PARHI, D. R. (2013). Controlling the motion of an autonomous mobile robot using various techniques : a review. *Journal of Advance Mechanical Engineering*, 1:24–39.
- [Monmarché, 2000] MONMARCHÉ, N. (2000). *Algorithmes de fourmis artificielles : applications à la classification et à l'optimisation*. Thèse de doctorat, Université François Rabelais.
- [Monmarché *et al.*, 2001] MONMARCHÉ, N., SLIMANE, M. et VENTURINI, G. (2001). L'algorithme antclass : classification non supervisée par une colonie de fourmis artificielles. *Extraction des Connaissances et Apprentissage : Apprentissage et évolution*, 1(3):131–166.
- [Morelli *et al.*, 2006] MORELLI, C., NICOLI, M., RAMPA, V., SPAGNOLINI, U. et ALIPPI, C. (2006). Particle filters for rss-based localization in wireless sensor networks : An experimental study. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 4, pages IV–IV. IEEE.
- [Moritz, 1980] MORITZ, H. (1980). Geodetic reference system 1980. *Journal of Geodesy*, 54(3):395–405.
- [Nadimi *et al.*, 2007] NADIMI, E. S., SØGAARD, H. T., OUDSHOORN, F. W., BLANES-VIDAL, V. et BAK, T. (2007). Monitoring cow behavior parameters based on received signal strength using wireless sensor networks. *Research Centre Bygholm, Horsens, Denmark*, 3:95–102.
- [Nagy et Farkas, 2000] NAGY, L. et FARKAS, L. (2000). Indoor base station location optimization using genetic algorithms. In *Personal, Indoor and Mobile Radio Communications, 2000. PIMRC 2000. The 11th IEEE International Symposium on*, volume 2, pages 843–846. IEEE.
- [Neskovic *et al.*, 2000] NESKOVIC, A., NESKOVIC, N. et PAUNOVIC, G. (2000). Modern approaches in modeling of mobile radio systems propagation environment. *Communications Surveys & Tutorials, IEEE*, 3(3):2–12.
- [Newman et Girvan, 2004] NEWMAN, M. E. J. et GIRVAN, M. (2004). Finding and evaluating community structure in networks. *Phys. Rev. E*, 69:026113.
- [Nguyen et Luo, 2013] NGUYEN, K. et LUO, Z. (2013). Evaluation of bluetooth properties for indoor localisation. In *Progress in Location-Based Services*, pages 127–149. Springer.
- [Niculescu et Nath, 2003] NICULESCU, D. et NATH, B. (2003). Dv based positioning in ad hoc networks. *Telecommunication Systems*, 22(1):267–280.

- [Nokia, 2012] NOKIA (2012). Indoor location Alliance. <http://press.nokia.com/2012/08/23/accurate-mobile-indoor-positioning-industry-alliance-called-in-location-to-promote-deployment-of-location-based-indoor-services-and-solutions/>. [En Ligne; Consulté le 16 Août 2013].
- [OAuth, 2013] OAUTH (2013). OAuth 2.0 Framework. <http://oauth.net/>. [En ligne; Consulté le 16 Août 2013].
- [Observatoire Société, 2013] OBSERVATOIRE SOCIÉTÉ (2013). Centre d'observatoire de la société. <http://www.observationsociete.fr/handicap/>. [En ligne; Consulté le 16 Août 2013].
- [Opensocial Foundation, 2013] OPENSOCIAL FOUNDATION (2013). Opensocial. <http://opensocial.org>. [En ligne; Consulté le 16 Août 2013].
- [Oudin, 2008] OUDIN, H. (2008). *Méthode des éléments finis*. École des Mines de Nantes.
- [Pahlavan *et al.*, 2002] PAHLAVAN, K., LI, X. et MAKELA, J. (2002). Indoor geolocation science and technology. *Communications Magazine, IEEE*, 40(2):112–118.
- [Patwari *et al.*, 2005] PATWARI, N., ASH, J., KYPEROUNTAS, S., HERO III, A., MOSES, R. et CORREAL, N. (2005). Locating the nodes : cooperative localization in wireless sensor networks. *Signal Processing Magazine, IEEE*, 22(4):54–69.
- [Pei *et al.*, 2010] PEI, L., CHEN, R., LIU, J., KUUSNIEMI, H., TENHUNEN, T. et CHEN, Y. (2010). Using inquiry-based bluetooth rssi probability distributions for indoor positioning. *Journal of Global Positioning Systems*, 9(2):122–130.
- [Pelleg et Moore, 2000] PELLEG, D. et MOORE, A. W. (2000). X-means : Extending k-means with efficient estimation of the number of clusters. *In Proceedings of the Seventeenth International Conference on Machine Learning, ICML '00*, pages 727–734, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- [Peterson *et al.*, 2008] PETERSON, G. L., MAYER, C. B. et KUBLER, T. L. (2008). Ant clustering with locally weighted ant perception and diversified memory. *Swarm Intelligence*, 2(1):43–68.
- [Pfeifer, 2005] PFEIFER, T. (2005). Redundant positioning architecture. *Computer Communications*, 28(13):1575–1585.
- [Priyantha, 2005] PRIYANTHA, N. B. (2005). *The cricket indoor location system*. Thèse de doctorat, Massachusetts Institute of Technology.
- [Priyantha *et al.*, 2000] PRIYANTHA, N. B., CHAKRABORTY, A. et BALAKRISHNAN, H. (2000). The Cricket location-support system. *Proceedings of the 6th annual international conference on Mobile computing and networking - MobiCom '00*, pages 32–43.
- [Qubulus, 2013a] QUBULUS (2013a). Liste de systèmes de positionnement. <http://www.qubulus.com/the-indoor-positioning-market-shaped-list-of-indoor-positioning-companies/>. [En ligne; Consulté le 16 Août 2013].
- [Qubulus, 2013b] QUBULUS (2013b). Qubulus indoor positioning. <http://www.qubulus.com/>. [En ligne; Consulté le 16 Août 2013].

- [Rappaport, 2002] RAPPAPORT, T. S. (2002). *Wireless Communications : Principles and Practice (2nd Edition)*. Prentice Hall, 2 édition.
- [Rasmussen, 2000] RASMUSSEN, C. E. (2000). The Infinite Gaussian Mixture Model. *Processing*, 12(11):554–560.
- [Régnier, 1965] RÉGNIER, S. (1965). Sur quelques aspects mathématiques des problèmes de classification automatique. *Mathématiques et sciences humaines*, 4:175–191.
- [Sakamoto *et al.*, 1986] SAKAMOTO, Y., ISHIGURO, M. et KITAGAWA, G. (1986). Akaike information criterion statistics. *Dordrecht, The Netherlands : D. Reidel*.
- [Santinelli *et al.*, 2009] SANTINELLI, G., GIGLIETTI, R. et MOSCHITTA, A. (2009). Self-calibrating indoor positioning system based on zigbee® devices. *In Instrumentation and Measurement Technology Conference, 2009. I2MTC'09. IEEE*, pages 1205–1210. IEEE.
- [Sarkar *et al.*, 2003] SARKAR, T., JI, Z., KIM, K., MEDOURI, A. et SALAZAR-PALMA, M. (2003). A survey of various propagation models for mobile communication. *Antennas and Propagation Magazine, IEEE*, 45(3):51–82.
- [Sasaki, 2007] SASAKI, Y. (2007). The truth of the f-measure. *Teach Tutor mater*, pages 1–5.
- [Satoh, 2012] SATOH, I. (2012). Building context-aware services from non-context-aware services. *Ambient Intelligence-Software and Applications*, 153:59–66.
- [Savarese *et al.*, 2001] SAVARESE, C., RABAEY, J. et BEUTEL, J. (2001). Location in distributed ad-hoc wireless sensor networks. *In Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, volume 4, pages 2037–2040. IEEE.
- [Savvides *et al.*, 2001] SAVVIDES, A., HAN, C. et STRIVASTAVA, M. (2001). Dynamic fine-grained localization in ad-hoc networks of sensors. *In International Conference on Mobile Computing and Networking : Proceedings of the 7 th annual international conference on Mobile computing and networking*, volume 2001, pages 166–179.
- [Schilit *et al.*, 1994] SCHILIT, B., ADAMS, N. et WANT, R. (1994). Context-aware computing applications. *In Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on*, pages 85–90. IEEE.
- [Schiller et Voisard, 2004] SCHILLER, J. et VOISARD, A. (2004). *Location-based services*. Morgan Kaufmann Pub.
- [Schmidt-Dannert et SNET, 2010] SCHMIDT-DANNERT, A. et SNET, S. (2010). Positioning technologies and mechanisms for mobile devices. *In Seminar Master Module SNET2 (TU-Berlin, ed.)*.
- [Schölkopf *et al.*, 1998] SCHÖLKOPF, B., SMOLA, A. et MÜLLER, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.*, 10(5):1299–1319.
- [Schrader *et al.*, 2010] SCHRADER, A., CARLSON, D. et ROTHENPIELER, P. (2010). Smartassist-wireless sensor networks for unobtrusive health monitoring. *In Proceedings of the 5th Behaviour Monitoring and Interpretation Workshop at the 33rd German Conference on Artificial Intelligence*.

- [Schwarz, 1978] SCHWARZ, G. (1978). Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464.
- [Schwarz, 2013] SCHWARZ, R. . (2013). Near Field Communication (NFC) Technology and Measurements.
- [Sendova-Franks et Franks, 1999] SENDOVA-FRANKS, A. B. et FRANKS, N. R. (1999). Self-assembly, self-organization and division of labour. *Philosophical Transactions of the Royal Society of London. Series B : Biological Sciences*, 354(1388):1395–1405.
- [Simek et al., 2013] SIMEK, M., POKORNY, J., BOTTA, M. et MRAZ, L. (2013). Handheld analyzer of iee 802.15. 4 phy and mac frames. *In Ad Hoc Networks*, pages 17–28. Springer.
- [Simon et Alouini, 2004] SIMON, M. et ALOUINI, M. (2004). *Digital communication over fading channels*, volume 86. Wiley-IEEE Press.
- [SM4ALL Project, 2013] SM4ALL PROJECT (2013). Smart Homes for All. <http://www.sm4all-project.eu/>. [En ligne ; Consulté le 16 Août 2013].
- [Stallings et Brown, 2008] STALLINGS, W. et BROWN, L. (2008). *Computer security*. Prentice-Hall.
- [Stansfield, 1947] STANSFIELD, R. (1947). Statistical theory of df fixing. *Electrical Engineers-Part IIIA : Radiocommunication, Journal of the Institution of*, 94(15):762–770.
- [Steg et al., 2006] STEG, H., STRESE, H., LOROFF, C., HULL, J. et SCHMIDT, S. (2006). Europe is facing a demographic challenge ambient assisted living offers solutions. *IST project report on ambient assisted living*.
- [Sun et al., 2009] SUN, H., DE FLORIO, V., GUI, N. et BLONDIA, C. (2009). Promises and challenges of ambient assisted living systems. *In Information Technology : New Generations, 2009. ITNG'09. Sixth International Conference on*, pages 1201–1207. IEEE.
- [Taskar et al., 2001] TASKAR, B., SEGAL, E. et KOLLER, D. (2001). Probabilistic classification and clustering in relational data. *In IJCAI*, pages 870–878.
- [TI, 2013] TI (2013). Texas instruments. <http://e2e.ti.com/>. [En ligne ; Consulté le 16 Août 2013].
- [Tibshirani et al., 2000] TIBSHIRANI, R., WALTHER, G. et HASTIE, T. (2000). Estimating the Number of Clusters in a Dataset via the Gap Statistic. *Journal of the Royal Statistical Society, Series B*, 63:411–423.
- [Torrieri, 1984] TORRIERI, D. J. (1984). Statistical Theory of Passive Location Systems. *IEEE Transactions On Aerospace And Electronic Systems*, AES-20(2):183–198.
- [Umeyama, 1988] UMEYAMA, S. (1988). An eigendecomposition approach to weighted graph matching problems. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(5):695–703.
- [Valaee et Li, 2012] VALAEE, S. et LI, X. (2012). Recent advances in wireless localization technologies. *International Journal of Wireless Information Networks*, 19:1–3.
- [Wallace et Boulton, 1968] WALLACE, C. S. et BOULTON, D. M. (1968). An information measure for classification. *Computer Journal*, 11(2):185–194.

## BIBLIOGRAPHIE

---

- [Yan, 2010] YAN, J. (2010). *Algorithms for Indoor Positioning Systems Using Ultra-Wideband Signals*. Thèse de doctorat, Delft University of Technology.
- [Zaharie et Zamfirache, 2005] ZAHARIE, D. et ZAMFIRACHE, F. (2005). Dealing with noise in ant-based clustering. *In Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 3, pages 2395–2401. IEEE.
- [Zhao, 2002] ZHAO, Y. (2002). Standardization of mobile phone positioning for 3g systems. *Communications Magazine, IEEE*, 40(7):108–116.
- [Zirari et al., 2010] ZIRARI, S., CANALDA, P. et SPIES, F. (2010). Wifi gps based combined positioning algorithm. *In Wireless Communications, Networking and Information Security (WCNIS), 2010 IEEE International Conference on*, pages 684–688. IEEE.



## Résumé :

Le concept du « smart » envahit de plus en plus notre vie quotidienne. L'exemple type est sans doute le smartphone. Celui-ci est devenu au fil des ans un appareil incontournable. Bientôt, c'est la ville, la voiture, la maison qui seront « intelligentes ».

L'intelligence se manifeste par une capacité d'interaction et de prise de décision entre l'environnement et l'utilisateur. Ceci nécessite des informations sur les changements d'états survenus des deux côtés. Les réseaux de capteurs permettent de collecter ces données, de leur appliquer des pré-traitements et de les transmettre aux applications. Ces réseaux de par certaines de leurs caractéristiques se rapprochent de l'intelligence collective, dans le sens, où des entités de faibles capacités se coordonnent automatiquement, sans intervention humaine, de façon décentralisée et distribuée pour accomplir des tâches complexes. Ces méthodes bio-inspirées ont servi à la résolution de plusieurs problèmes, surtout l'optimisation, ce qui nous a encouragé à étudier la possibilité de les utiliser pour les problèmes liés à l'*Ambient Assisted Living* ou AAL et à la classification automatique de données.

L'AAL est un sous-domaine des services dits basés sur le contexte, et a pour objectifs de faciliter la vie des personnes âgées et handicapées dans leurs défis quotidiens. Pour ce faire, il détermine le contexte et, sur cette base, propose divers services. Deux éléments du contexte nous ont intéressés : le handicap et la position.

Bien que la détermination de la position (localisation, positionnement) se fasse à l'extérieur des bâtiments avec des précisions très satisfaisantes, elle rencontre plusieurs difficultés à l'intérieur des bâtiments, liées à la propagation des ondes électromagnétiques dans les milieux difficiles, aux coûts des systèmes, à l'interopérabilité, etc.

Nos travaux se sont intéressés au positionnement des personnes handicapées à l'intérieur de bâtiments en utilisant un réseau de capteurs afin de déterminer les caractéristiques de l'onde électromagnétique (puissance, temps, angle) pour estimer la position par méthodes géométriques (triangulation, latération), méthodes de fingerprinting (k plus proches voisins), par des filtres bayésiens (filtre de Kalman). L'application est d'offrir des services types AAL tel que la navigation. Nous avons élargi la notion de réseau de capteurs pour prendre en compte tout appareil capable d'émettre et de recevoir une onde électromagnétique et se trouvant dans l'environnement. Nous avons aussi appliqué l'algorithme API sur la classification automatique de données. Enfin, nous avons proposé une architecture à middleware pour la localisation *indoor*.

**Mots clés :** Réseaux de capteurs, Intelligence Collective, Clustering, Positionnement *indoor*, Ambient Assisted Living, Personnes âgées et handicapées.

## Abstract :

The concept of « smart » invades more and more our daily life. A typical example is the smartphone, which becomes by years an essential device. Soon, it's the city, the car and the home which will become « smart ». The intelligence is manifested by the ability for the environment to interact and to take decisions in its relationships with users and other environments. This needs information on state changes occurred on both sides. Sensor networks allow to collect these data, to apply on them some pre-processings and to transmit them. Sensor network, towards some of their characteristics are closed to Swarm Intelligence in the sense that small entities with reduced capabilities can cooperate automatically, in unattended, decentralised and distributed manner in order to accomplish complex tasks. These bio-inspired methods have served as basis for the resolution of many problems, mostly optimization and this inspired us to apply them on problems met in Ambient Assisted Living and on the data clustering problem.

AAL is a sub-field of context-aware services, and its goals are to facilitate the everyday life of elderly and disable people. These systems determine the context and then propose different kind of services. We have used two important elements of the context : the position and the disability. Although positioning has very good precision outdoor, it faces many challenges in indoor environments due to the electromagnetic wave propagation in harsh conditions, the cost of systems, interoperability, etc.

Our works have been involved in positioning disabled people in indoor environment by using wireless sensor network for determining the characteristics of the electromagnetic wave (signal strength, time, angle) for estimating the position by geometric methods (triangulation, lateration), fingerprinting methods (k-nearest neighbours), bayesian filters (Kalman filter). The application is to offer AAL services like navigation. Therefore we extend the definition of sensor node to take into account any device, in the environment, capable of emitting and receiving a signal. Also, we have studied the possibility of using *Pachycondylla Apicalis* for data clustering and for indoor localization by casting this last problem as data clustering problem. Finally we have proposed a system based on a middleware architecture.

**Keywords :** Wireless Sensor Networks, Swarm intelligence, Localization, Positioning, Data Clustering, Ambient Assisted Living, Context-Aware services, Location-based services, Disabled and elderly people.