

N° d'ordre : 09/2014-M/INF

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'enseignement supérieur et de la recherche Scientifique
Université des sciences et de la Technologie Houari Boumediène

Faculté de l'Electronique et d'Informatique



MEMOIRE

Présenté pour l'obtention du **diplôme** de **MAGISTER**

En : INFORMATIQUE

Spécialité : Informatique Mobile

Par : Sid Ali HASNI

Sujet

Les data-ferries dans les réseaux de capteurs sans fil.

Soutenu publiquement, le 18/12/2014, devant le jury composé de :

M.	Youcef ZAFOUNE	Maître de conférences/A	à l'USTHB	Président
Mme.	Samira MOUSSAOUI	Professeur	à l'USTHB	Directrice de mémoire
M.	Mohamed AISSANI	Maître de conférences/A	à l'EMP	Examineur
M.	Mustapha-Rédha SENOUCI	Maître de conférences/B	à l'EMP	Invité

Remerciements

Louange à notre Seigneur 'ALLAH' qui m'a doté de la merveilleuse faculté de raisonnement. Louange à notre Créateur qui m'a incité à acquérir le savoir. C'est à lui que s'adresse toute ma gratitude en premier lieu.

Je présente mes remerciements les plus sincères à ma directrice de mémoire, Mme Samira MOUSSAOUI, professeur à l'Université des Sciences et de la Technologie Houari Boumediene (USTHB), et à mon co-directeur de mémoire, Mr Redha mustapha SENOUCI, Maître de conférences /B à l'Ecole Militaire Polytechnique (EMP), pour leurs disponibilité et orientations le long de mon travail.

Je tiens à remercier également Mr. Y. ZAFOUNE, Maître de conférences /A à l'Université des Sciences et de la Technologie Houari Boumediene (USTHB) et Mr Mohamed AISSANI, Maître de conférences /A à l'Ecole Militaire Polytechnique (EMP), d'avoir accepté de juger ce travail.

Mes remerciements s'adressent aussi à mes amis de l'Ecole Militaire Polytechnique (EMP), particulièrement ceux attachés au centre de documentation, qui m'ont aidé.

Un remerciement particulier à mon camarade Mr Taha BENSIRADJ et à mon ancien chef de centre de documentation Mr Nouredine MAIZA, qui m'ont beaucoup aidé à finaliser ce travail et avec lesquels j'ai passé des bons moments. Je les souhaite à mon tour une bonne continuation.

Finalement je remercie chaleureusement mon père, ma mère, ma femme, mes frères, ma sœur et tous les membres de ma famille pour la confiance qu'ils m'accordent, leur amour, et leurs encouragements.

Dédicace

À mon père et mon grand père ;

À ma mère et ma grand-mère ;

À ma femme ;

À mon petit Abdallah ;

À mes frères et ma sœur ;

À tous mes amis ;

À tous ceux qui me sont chers...

SOMMAIRE

Introduction générale	1
Chapitre 1 Généralités sur les réseaux de capteurs	3
1.1. Introduction	3
1.2. Les réseaux de capteurs sans fil	3
1.2.1. Définition d'un capteur sans fil	4
1.2.2. Description des réseaux de capteur sans fil	6
1.3. Les réseaux de capteurs sans fil mobiles	7
1.3.1. Architecture d'un RCSF mobile	8
1.3.2. Avantages de la mobilité	10
1.4. Les data ferries	12
1.4.1. Architecture à trois niveaux des data ferries	13
1.4.2. La collecte de données par les data ferries	14
1.4.3. Modes pour la collecte des données	15
1.4.4. Exemples d'application des data ferries	17
1.4.4.1. Les drones ou UAVs (<i>Unmanned Aerial Vehicles</i>) :	17
1.4.4.2. Les véhicules sous-marins autonomes ou AUVs (<i>Autonomous Underwater Vehicles</i>) :	18
1.5. Conclusion	19
Chapitre 2 La planification de la trajectoire des data ferries	20
2.1. Introduction	20
2.2. Types de la mobilité des data ferries	21
2.3. Classification des approches de la planification de la trajectoire des data ferries ..	22
2.3.1. La collecte de données par communication directe (Direct-Contact Data Collection)	22
2.3.1.1. Trajectoire stochastique de collecte de données (Stochastic Data Collection Trajectory) ..	23
2.3.1.2. Chemin TSP pour la collecte de données (TSP tour for data collection)	23
2.3.1.3. Chemin Label-Covering pour la collecte de données (Label-Covering tour for Data collection)	24
2.3.2. La collecte de données basée points de rendez-vous (Rendezvous-Based Data Collection)	25
2.3.2.1. La sélection des RPs par chemin fixe	25
2.3.2.2. La sélection des RPs basée arbre	26
2.3.2.3. La sélection des RPs par Clustering	27
2.4. La planification de la trajectoire basée TSP	28
2.4.1. La classe des travaux TSP	28
2.4.1.1. La planification d'ordonnancement	29
2.4.1.2. La planification adaptative	32
2.4.1.3. La planification Single-hop	37
2.4.2. La classe des travaux TSPN	40
2.4.2.1. TSPN avec problème d'optimisation	41
2.4.2.2. TSPN avec champ de communication en intersection	42
2.4.3. Synthèse :	45
2.5. Conclusion	46

Chapitre 3 PSFPP : La solution pour la planification de la trajectoire de data ferry proposée	47
3.1. Introduction	47
3.2. Formulation du problème de la planification de la trajectoire de data ferry	48
3.2.1. La formulation mathématique de problème TSPN :	48
3.2.1.1. Optimisation énergétique :	49
3.2.1.2. Optimisation de la trajectoire :	49
3.3. Modèle de connectivité probabiliste et les coûts de communication	50
3.4. Nouvelle formulation du problème de planification de la trajectoire de data ferry	52
3.4.1. Modèle réseau et hypothèses	53
3.4.2. Formulation du problème	54
3.5. Notre approche de résolution	54
3.5.1. Définition de la fonction score	55
3.5.1.1. Étape 1 : Construction des vecteurs de couverture	56
3.5.1.2. Étape 2 : Construction de la matrice de couverture de la grille « MCG »	57
3.5.1.3. Étape 3 : Détermination des sous-ensembles de couverture	57
3.5.1.4. Étape 4 : Détermination de la longueur de chemin de chaque sous-ensemble	59
3.5.1.5. Étape 5 : Définition de la fonction objectif	60
3.6. Conclusion	62
 Chapitre 4 Implémentation et évaluation de la solution PSFPP proposée	 63
4.1. Introduction	63
4.2. Les heuristiques TSP et SSP	63
4.2.1. Le problème de voyageur de commerce TSP	63
4.2.2. Le SSP (Sub Set Sum Problem)	66
4.3. Etude expérimentale du modèle de communication probabiliste	67
4.4. Evaluation de la trajectoire poursuivie par le data ferry	68
4.4.1. Implémentation	68
4.4.2. Etude de la trajectoire de data ferry	68
4.5. Evaluation de la solution proposée	71
4.5.1. Environnement de simulation	71
4.5.2. Paramètres de simulation	72
4.5.3. Paramètres d'étude	73
4.5.4. Métrique à mesurer	73
4.5.5. Scénarios de simulation	74
4.5.5.1. Scénario 1 : Variation de la distance seuil ds	74
4.5.5.2. Scénario 2 : Variation de la densité des nœuds	74
4.5.6. Synthèse des résultats de simulation	75
4.6. Conclusion	76
 Conclusion générale et perspectives	 77
 Références bibliographiques	 78

Liste des figures

Figure 1.1. Exemple de capteur sans fil MICAZ de Crossbow.	4
Figure 1.2. Architecture physique d'un capteur.	6
Figure 1.3. Schéma général d'un réseau de capteurs.	6
Figure 1.4. Exemples des nœuds capteurs mobiles.	8
Figure 1.5. Architecture plate des RCSFs mobiles.	8
Figure 1.6. Architecture 2-tiers des RCSFs mobiles.	9
Figure 1.7. Architecture 3-tiers des RCSFs mobiles.	10
Figure 1.8. Architecture trois tiers des data ferries	12
Figure 1.9. Scénario du processus de la collecte de données	15
Figure 1.10. La collecte de données à un seul saut.	16
Figure 1.11. La collecte de données à multi-saut.	16
Figure 1.12. Exemple sur les drones.	17
Figure 1.13. La surveillance d'un pont par les data ferries.	18
Figure 1.14. Exemple d'un véhicule sous-marin autonome.	18
Figure 1.15. La surveillance sous-marin par les data ferries	18
Figure 2.1. Types de la mobilité des data ferries.	21
Figure 2.2. Taxonomie des approches de la planification de la trajectoire des data ferries.	22
Figure 2.3. La différence entre les chemins TSP et TSPN.	24
Figure 2.4. Le chemin Label-Covering	25
Figure 2.5. La sélection des RPs par chemin fixe.	26
Figure 2.6. La sélection des RPs basé arbre	27
Figure 2.7. La sélection des RPs par Clustering	28
Figure 2.8. Nouvelle classification des approches de la collecte de données de la classe TSP.	29
Figure 2.9. La planification de la trajectoire par l'algorithme PBS.	31
Figure 2.10. La transmission multi-saut pour le transport des messages urgents.	32
Figure 2.11. La décomposition de la trajectoire en segments	33
Figure 2.12. Planification du chemin (exemples pour les data ferries)	34
Figure 2.13. Le chemin de data ferry à travers les polling points.	38

Figure 2.14. La planification de la trajectoire par EM-TSP.	39
Figure 2.15. L'heuristique de EM-TSP pour la planification de la trajectoire.	39
Figure 2.16. L'heuristique pour la résolution de problème TSPN.....	41
Figure 2.17. La planification TSPN avec des champs de communication de taille différente.....	42
Figure 2.18. La planification de la trajectoire par TspnTour.	43
Figure 2.19. La planification de la trajectoire par SparseTspnTour.	43
Figure 3.1. Planification de la trajectoire de data ferry.....	48
Figure 3.2. Le modèle de communication binaire.	50
Figure 3.3. Modèle de propagation avec $\gamma = 2$, $\sigma_{\epsilon} = 4$, $P_t = 0$ dBm.....	51
Figure 3.4. Le modèle de connectivité.	52
Figure 3.5. Une modélisation en grille bidimensionnelle.....	53
Figure 3.6. illustration schématique de la fonction score.	55
Figure 3.7. Formes de l'ensemble des solutions optimales au sens de Pareto	61
Figure 4.1.Fonctionnement de l'algorithme d'insertion.	64
Figure 4.2.Fonctionnement de l'algorithme de la plus proche insertion.	65
Figure 4.3.Comparaison entre la puissance reçue mesurée et la puissance reçue théorique.	67
Figure4.4.La trajectoire du data ferry avec $\alpha > \beta$	69
Figure 4.5.La trajectoire du data ferry avec $\alpha < \beta$	70
Figure 4.6.La trajectoire du data ferry avec $\alpha = \beta$	70
Figure 4.7.Processus de simulation.....	72
Figure 4.8.Impact de la distance seuil sur les caractéristiques de la solution.....	74
Figure 4.9.Impact de la variation de la densité sur les caractéristiques de la solution...	75

Liste des tableaux

Tableau 2.1 .Récapitulatif des approches de la collecte de données basées TSP	44
Tableau 3.1. La forme du tableau des solutions.....	60
Tableau 4.1. Les paramètres utilisés pour la simulation.....	73

Glossaire

AP	Access Point
ASD	Adaptative Speed Control
AUV	Autonomous Underwater Vehicle
ds	Distance Seuil
DTN	Delay Tolerant Network
GPRS	Genral Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile communications
IN	Intermediate Navigator
MANET	Mobile Ad hoc Network
MCG	Matrice de Couverture de la grille
MWSN	Mobile Wireless Sensor Network
NA	Navigation Agent
PDA	Personal Digital Assistant
RCSF	Réseau de Capteurs Sans Fil
Rc	Rayon de Communication
RSSI	Received Signal Strength Indicator
SB	Station de Base
SCD	Stop to Collect Data
SSP	Subset Sum Problem
TinyOS	Tiny Operating System
TSP	Traveling Salesman Problem
TSPN	Traveling Salesman Problem with Neighborhood
UAV	Unmanned Aerial Vehicle
VCC	Vecteur de Couverture du Collecteur
WSN	Wireless Sensor Network

Introduction générale

Les réseaux de capteurs sans fil (RCSF ou Wireless sensor networks WSN) sont une catégorie particulière des réseaux ad-hoc comprenant des nœuds micro-capteur fonctionnant avec de faibles capacités de calcul et de stockage. Ces nœuds récoltent les données d'intérêt dans le champ d'intérêt et les transmettent, via des canaux sans fil, à un récepteur ou une station de base (sink), généralement située à l'extérieur du champ d'intérêt (champ de captage). Les RCSFs comprennent des centaines voire des milliers de nœuds capteurs et sont généralement déployés à grande échelle dans des environnements non structurés qui sont difficiles à contrôler manuellement. Il existe différentes applications typiques des réseaux de capteurs telle que la surveillance environnementale, la détection des incendies de forêts, la surveillance d'un champ de bataille, le contrôle de la circulation, le suivi des patients, la poursuite de cibles, etc.

Ces réseaux, du fait de leurs fortes contraintes (énergétique, mémoire et puissance de calcul), soulèvent de nouveaux challenges dans le domaine des communications radio. La communauté scientifique a commencé à se pencher sur la gestion de la mobilité au sein d'un même réseau de capteurs. La mobilité peut être envisagée comme la source de nouvelles solutions face à des problèmes déjà identifiés, mais elle soulève également de nouveaux défis.

L'utilisation des nœuds mobiles en particulier les « data-ferries » pour la collecte des données dans les réseaux de capteurs peut représenter un grand avantage, pour ces réseaux. En effet, cette approche de collecte permet d'avoir des systèmes de surveillance avec des durées de vie optimisées, un overhead réduit et un risque limité de congestion en trafic.

De toute évidence, la planification de la trajectoire de ces entités mobiles « data ferries » a un impact significatif sur le schéma optimal de la collecte de données. Le problème de la planification de la trajectoire de data ferry peut être formulé par un problème de la classe TSP (Traveling Salesman Problem). La formulation classique consiste en un ensemble de nœuds statiques déployés dans une zone d'intérêt et une entité mobile qui se déplace dans la scène pour la collecte de données. Le but de la planification est d'optimiser certains paramètres de réseau tel que : la latence, l'énergie, le temps de la collecte de donnée, etc [MAR11, ANA08, MUN07].

Dans la littérature, les approches de la planification de la trajectoire, avec considération du champ de communication, modélisent le champ de communication par une forme circulaire appelée communément le modèle de communication binaire ou le modèle disque. Ce modèle permet de ramener le problème de la planification de la trajectoire de data ferry à un problème géométrique. Bien que le modèle disque simplifié l'analyse, il est considéré comme étant trop simpliste et ne reflète pas fidèlement la réalité. Dans la réalité et de point de vue géométrique le champ de communication n'est pas réellement un disque, il n'y a pas une frontière nette entre les communications réussies et échouées. En se basant sur cette constatation, dans ce mémoire, nous avons considéré une nouvelle formulation de ce problème par la prise en compte d'un modèle réel de communication des capteurs. Selon nos connaissances de l'état de l'art sur cette question, cet aspect n'a pas été considéré au préalable. Pour notre proposition, nous définissons un modèle de prédiction des coûts de communication.

Dans cette étude, nous proposons une nouvelle formulation de problème de la planification de la trajectoire de data ferry qui incorpore un modèle de communication réaliste. Pour résoudre ce nouveau problème, nous proposons une solution basée sur un modèle réseau simple et discret, et la résolution est effectuée en étapes. Dans chaque étape, de nouveaux concepts sont introduits et des heuristiques de la classe TSP et SSP (Subset Sum Problem) [GER94, KUM13] sont appliquées.

Le présent mémoire est organisé en quatre chapitres. Dans le chapitre 1, nous présentons quelques généralités sur les RCSFs, les RCSFs mobiles et les data ferries, et nous nous focalisons particulièrement sur le concept des data ferries dans ce type de réseau. Dans le chapitre 2, nous proposons une taxonomie des approches de la collecte de données par les data ferries. Ensuite, nous présentons les approches de la planification de la trajectoire basées TSP et nous présentons une nouvelle classification. A la fin de ce chapitre, nous faisons une synthèse sur les approches de la collecte de données étudiées. Dans le chapitre 3, nous proposons une nouvelle solution pour la planification de la trajectoire de data ferry. Pour cela, nous commençons par développer un nouveau modèle de communication probabiliste. Ensuite, nous reformulons le problème de la planification en considérant ce nouveau modèle. A la fin, nous proposons une solution à ce nouveau problème. Dans le chapitre 4, nous effectuons une évaluation de la solution proposée par un ensemble de tests. Le mémoire s'achève par une conclusion et quelques perspectives.

Chapitre 1 Généralités sur les réseaux de capteurs

1.1. Introduction

Actuellement, la mobilité est devenue un domaine de recherche très important pour la communauté des RCSFs. Bien que, le déploiement des réseaux de capteurs (RCSFs) n'ait jamais été envisagé comme étant entièrement statique, la mobilité a avant tout plusieurs challenges à surmonter, en incluant la connectivité, la couverture, la consommation d'énergie et bien d'autres. Toutefois, les études récentes ont montré les bénéfices de la mobilité [WAN09, AMU09, GAN03].

Récemment, de manière similaire aux tendances de la recherche dans les réseaux mobiles ad hoc (MANET) et les réseaux tolérants aux délais (Delay Tolerant Networks, DTNs) [MAR11], la mobilité a également été introduite pour les réseaux de capteurs. En fait, la mobilité dans les réseaux de capteurs est utile pour plusieurs raisons : connectivité, réduction du coût, fiabilité et optimisation énergétique. Cette mobilité est assurée par l'utilisation des entités mobiles pour la collecte de données dans les réseaux de capteurs et en particulier par l'utilisation des data ferries [MAR11, EKI06, BOR10, HAR09].

Ce chapitre commence en premier lieu par donner un petit aperçu sur les réseaux de capteurs sans fil. Ensuite, une étude de la mobilité dans les réseaux de capteurs est entamée afin de spécifier l'apport de la mobilité. Enfin, nous décrivons l'approche de la collecte de données dans les réseaux de capteurs par l'utilisation des nœuds mobiles et, en particulier, les data ferries qui sont le sujet de notre étude où nous détaillons le principe de fonctionnement de ses environnements et des architectures utilisées.

1.2. Les réseaux de capteurs sans fil

Les développements réalisés ces dernières années dans les domaines des microsystèmes électromécaniques (Micro-electro-mechanical systems) ainsi que des techniques de communication sans fil ont permis de voir apparaître un nouveau type de réseaux, appelés Réseaux de Capteurs Sans Fil ou Wireless Sensor Networks (WSN) [KHE04].

Les RCSFs interconnectent des équipements miniatures capables d'observer leur environnement proche (température, niveau de pollution, informations médicales, etc.). L'utilisation d'un RCSF permet de remplir de nombreuses tâches comme le suivi des déplacements des troupes, les services d'aide aux personnes âgées ou la surveillance des paramètres de santé de patients hospitalisés [SEV06, CHO03].

1.2.1. Définition d'un capteur sans fil

Un capteur sans fil est un dispositif physique, de taille miniature, qui a pour but essentiel d'accomplir trois tâches complémentaires : le relevé d'une grandeur physique, le traitement éventuel de cette information, et la communication avec d'autres capteurs afin de transmettre cette information. Pour pouvoir accomplir ces tâches, le capteur est composé principalement de quatre unités de base : l'unité d'acquisition, l'unité de traitement, l'unité de communication, et l'unité de contrôle d'énergie [GER09]. Selon le domaine d'application, il peut aussi contenir des modules supplémentaires tels qu'un système de localisation (GPS), ou bien un système générateur d'énergie (cellule solaire). Quelques micro-capteurs, plus volumineux, sont dotés d'un système mobilisateur chargé de les déplacer en cas de nécessité.



Figure 1.1. Exemple de capteur sans fil MICAZ de Crossbow.

Généralement, un nœud capteur est composé de quatre unités principales qui sont données sur la figure 1.2 :

- L'unité d'acquisition : généralement composée de deux sous-unités : le récepteur et le transducteur (convertissant le signal du récepteur en signal électrique). Le capteur fournit des signaux analogiques, basés sur le phénomène observé, au convertisseur Analogique/Numérique. Ce dernier transforme ces

signaux en un signal numérique compréhensible par l'unité de traitement [BEY09].

- L'unité de traitement : Elle comprend un processeur généralement associé à une petite unité de stockage. Elle fonctionne à l'aide d'un système d'exploitation spécialement conçu pour les micro-capteurs (par exemple, TinyOS) [TIN03, HIL03]. Elle exécute les protocoles de communications qui permettent de faire « collaborer » le nœud avec les autres nœuds du réseau. Elle peut aussi analyser les données captées pour alléger la tâche du nœud puits.
- L'unité de communication : Elle effectue toutes les émissions et réceptions des données sur un milieu « sans-fil ». Elle peut être de type optique (comme dans les nœuds), ou de type radiofréquence.
 - Les communications de type optique sont robustes vis-à-vis des interférences électriques. Néanmoins, ne pouvant pas établir de liaisons à travers des obstacles, elles présentent l'inconvénient d'exiger une ligne de vue permanente entre les entités communicantes.
 - Les unités de communication de type radiofréquence comprennent des circuits de modulation, démodulation, filtrage et multiplexage ; ceci implique une augmentation de la complexité et du coût de production du micro-capteur.

Concevoir des unités de communication de type radiofréquence avec une faible consommation d'énergie est un défi car pour qu'un nœud ait une portée de communication suffisamment grande, il est nécessaire d'utiliser un signal assez puissant et donc une énergie consommée importante. L'alternative consistant à utiliser de longues antennes n'est pas possible à cause de la taille réduite des micro-capteurs.

- Unité de contrôle d'énergie : L'unité de contrôle d'énergie est un système essentiel. Elle doit répartir l'énergie disponible aux autres modules et unités, de manière optimale (par exemple en réduisant les dépenses inutiles et en mettant en veille les composants inactifs). Cette unité peut aussi gérer des systèmes de

rechargement d'énergie à partir de l'environnement via des cellules photovoltaïques par exemple.

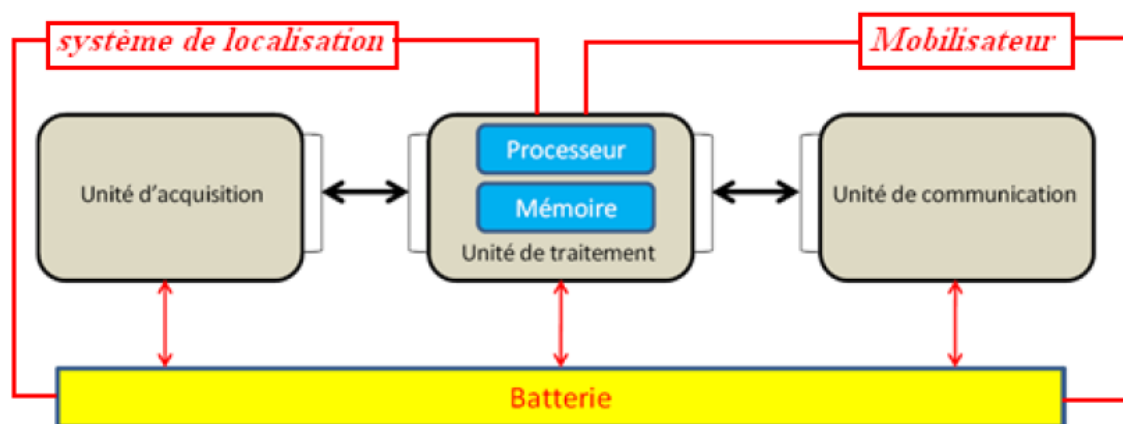


Figure 1.2. Architecture physique d'un capteur.

1.2.2. Description des réseaux de capteur sans fil

Les réseaux de capteurs forment une sous-classe des réseaux mobiles ad-hoc. Les nœuds capteurs, déployés en grand nombre, sont capables de récolter et de transmettre des données environnementales d'une manière autonome. La position de ces nœuds n'est pas obligatoirement prédéterminée. Ils sont dispersés aléatoirement à travers une zone géographique : appelée champ de capture. Le champ de capture définit le terrain d'intérêt pour le phénomène capturé. Cela permet un déploiement aléatoire dans des terrains inaccessibles ou dans des opérations de secours aux sinistrés.

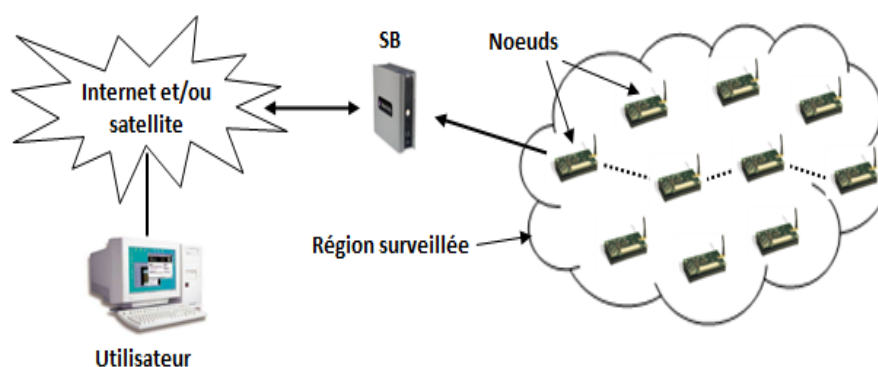


Figure 1.3. Schéma général d'un réseau de capteurs.

Généralement, les protocoles d'un RCSF doivent posséder des capacités d'auto-organisation. Une autre caractéristique unique des réseaux de capteurs est l'effort de

coopération des nœuds de capteurs. Les données capturées sont acheminées grâce à un routage multi-sauts vers un nœud considéré comme "point de collecte", appelé nœud collecteur (Station de Base ou Sink). Ce dernier peut être connecté à l'utilisateur du réseau via Internet ou un satellite. L'utilisateur peut ainsi adresser des requêtes aux nœuds du réseau en précisant le type de données requises ; les données environnementales capturées seront récoltées par le biais du collecteur [GER09].

Les nœuds de capteurs sont équipés d'un processeur embarqué. Au lieu d'envoyer des données brutes pour les nœuds responsables de la fusion, ils utilisent leurs capacités de traitement pour effectuer localement des calculs simples et transmettre les données nécessaires qui sont partiellement traitées [BOU07].

Chaque application a ses propres contraintes. Dans tous les domaines, le rôle d'un réseau de capteurs est cependant plus au moins le même. Les nœuds doivent surveiller certains phénomènes grâce à leurs capteurs puis ils envoient les informations capturées à un collecteur. Le collecteur est un nœud particulier doté généralement d'une puissance de calcul supérieure et d'une quantité d'énergie potentiellement infinie. Ce collecteur peut être connecté à Internet ou possède un lien radio de type GSM ou GPRS qui lui permet d'envoyer les informations (données ou alertes) à un centre de contrôle pour l'utilisateur final. Il peut y avoir plusieurs collecteurs mobiles ou fixes dans un réseau mais pour des raisons de coût, il y a beaucoup moins de collecteurs que de nœuds ordinaires.

1.3. Les réseaux de capteurs sans fil mobiles

Ces dix dernières années, les RCSFs ont connu une évolution remarquable au niveau de la construction du hardware, des protocoles de communication, du rendement des ressources et de bien d'autres aspects. Typiquement, les applications des RCSFs sont celles qui nécessitent l'observation de quelques phénomènes physiques à travers des échantillons de l'environnement surveillé. Souvent, les RCSFs sont déployés dans une zone géographique étendue. Une grande surface requiert un grand nombre de nœuds capteurs, ce qui implique un coût prohibitif et des interférences radio excessives. Une autre approche pour résoudre ce problème est d'employer des nœuds mobiles [LIU05, KOU11].

Généralement, un réseau de capteurs sans fil mobiles (MWSN : Mobile Wireless Sensor Network) est constitué de nœuds capteurs mobiles. Les RCSFs mobiles qui se composent à la fois des nœuds statiques et des nœuds mobiles, sont appelés RCSF hybrides, mais ils rentrent toujours dans les RCSFs mobiles [REN06]. La figure 1.4 représente des exemples de nœuds capteurs mobiles.



Figure 1.4. Exemples des nœuds capteurs mobiles.

Dans cette section, nous présentons une brève classification des RCSFs mobiles, en incluant les caractéristiques les distinguant des RCSFs statiques et les avantages liés à l'introduction de la mobilité des nœuds capteurs.

1.3.1. Architecture d'un RCSF mobile

Les réseaux de capteurs mobiles peuvent être organisés par l'architecture plate (flat), 2-tiers, ou 3-tiers hiérarchique [MUN07].

Dans l'architecture réseau plate (planaire) (figure 1.5), le réseau comprend un ensemble de dispositifs hétérogènes communicants en mode Ad hoc. Les dispositifs peuvent être mobiles ou statiques, mais ils communiquent tous au sein du même réseau. Des systèmes de navigation de base ont une architecture plate.

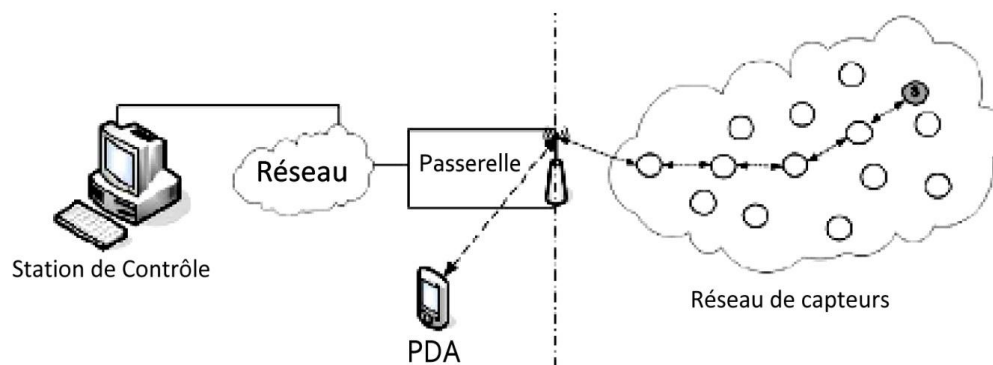


Figure 1.5. Architecture plate des RCSFs mobiles.

L'architecture deux-tiers (two-tier) (figure 1.6) consiste en un ensemble de nœuds statiques, et un ensemble de nœuds mobiles. Les nœuds mobiles forment un réseau recouvert pour aider à transmettre les données dans le réseau. Le réseau couvert peut inclure des dispositifs mobiles qui ont une grande capacité de traitement, une longue portée de communication, et une large bande passante. De plus, la densité du réseau couvert est satisfaite lorsque tous les nœuds sont toujours connectés, sinon le réseau devient disjoint. Dans ce cas, les entités mobiles se positionnent de manière afin de rétablir la connexion, toute en assurant l'arrivée à leur destination les paquets du réseau. Le système NavMote [FAN05] utilise cette approche.

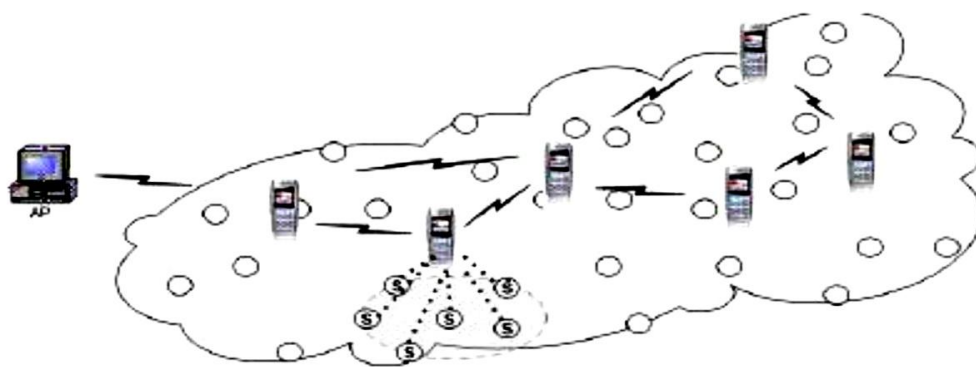


Figure 1.6. Architecture 2-tiers des RCSFs mobiles.

Dans l'architecture trois-tiers (three-tier) (figure 1.7), l'ensemble des nœuds statiques passe les données à l'ensemble des nœuds mobiles, qui à son tour les transmet à l'ensemble des points d'accès. Ce réseau hétérogène est destiné à couvrir des surfaces larges et être compatible avec plusieurs applications simultanément. Par exemple, on considère une application du réseau de capteur qui surveille un parking pour la disponibilité des places. Le capteur (premier tiers) diffuse (Broadcasts) les mises-à-jours de la disponibilité aux dispositifs mobiles (deuxième tiers), tel un téléphone portable ou PDA. A son tour, le téléphone portable transmet cette donnée de disponibilité aux points d'accès (troisième tiers), telle une station de base, et les données sont surchargées dans un serveur de base de données centralisée. Ensuite, les utilisateurs qui veulent localiser une place libre dans le parking peuvent accéder à la base de données.

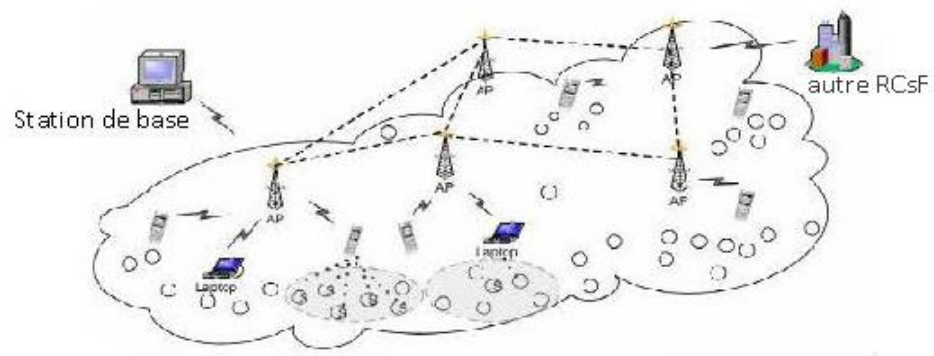


Figure 1.7. Architecture 3-tiers des RCSFs mobiles.

Au niveau du nœud, le capteur sans fil mobile se classe selon son rôle dans le réseau :

- ✓ **Capteur Intégré Mobile** (Mobile Embedded Sensor) : les nœuds intégrés mobiles ne contrôlent pas leurs propre mouvement ; par contre, leurs mouvement est dirigé par quelques forces externes, on utilise ce type par exemple, pour le fixer sur un animal ou l'attacher sur un bateau.
- ✓ **Capteur motivé mobile** (Mobile Actuated Sensor) : les nœuds peuvent aussi avoir la capacité de contrôler leurs propres mouvement, qui leurs permet de parcourir toute la région de sensation ; avec ce type de mobilité contrôlée, la spécification du déploiement peut être plus exacte, la couverture peut être maximisée, et les phénomènes spécifiques peuvent être ciblés et suivis.
- ✓ **Data ferry** : souvent, les capteurs n'ont pas besoin d'être mobiles, mais ils peuvent avoir un dispositif mobile afin de leurs collecter les données et les livrer à la station de base. Ce type d'entités mobiles est référé comme data ferries [SHA03]. Généralement ; on suppose que les data ferries peuvent recharger leur propre énergie automatiquement (points d'alimentation, énergie solaire,...etc).

1.3.2. Avantages de la mobilité

Les architectures traditionnelles des réseaux de capteurs reposent sur l'hypothèse que le réseau est dense, de sorte que n'importe quelle paire de nœuds du réseau puisse communiquer avec les autres par des chemins multi-sauts. En conséquence, dans la plupart des cas, les capteurs sont supposés être statiques, et la mobilité n'est pas considérée comme une option. Parallèlement aux tendances de la recherche dans les

réseaux mobiles ad hoc (MANET) et les réseaux tolérants aux délais (Delay Tolerant Networks, DTNs), la mobilité a également été introduite pour les réseaux de capteurs. En fait, la mobilité dans les réseaux de capteurs est utile pour plusieurs raisons [MAR11, NAT11] :

- **Connectivité** : Comme les nœuds sont mobiles, une architecture de déploiement dense des nœuds capteurs peut ne pas être exigée. En fait, les éléments mobiles peuvent établir des connexions entre des régions isolées, de sorte que les contraintes sur la connectivité réseau et le déploiement des nœuds capteurs puissent être réduites.
- **Coût** : Le coût dans un réseau de capteurs mobile est réduit, vu que moins de nœuds sont déployés. Bien que l'ajout des fonctionnalités de mobilité aux nœuds puisse être coûteux, dans de nombreux cas, il est possible d'exploiter les éléments mobiles qui sont déjà présents dans la zone de couverture (par exemple, les trains, les autobus, les navettes ou les voitures), et les attacher aux capteurs.
- **Fiabilité**. Les réseaux de capteurs traditionnels (statiques) sont denses et le paradigme de communication est souvent (ad hoc) multi-saut, donc dans cette architecture les interférences et les collisions agissent fortement sur la fiabilité. En plus, la perte des messages augmente avec le nombre de sauts, qui peut être assez élevé. Les éléments mobiles, au contraire, peuvent visiter les nœuds dans le réseau et collecter les données directement à travers une transmission à un seul saut. Cela ne réduit pas uniquement les collisions, mais aussi la perte des messages.
- **Optimisation énergétique** : Dans un réseau de capteurs sans fil les messages sont générés par les capteurs et collectés par le sink. Comme conséquence, les nœuds proche du sink sont plus surchargés que les autres, et sujets d'une déplétion d'énergie prématurée. Ce phénomène est appelé « *the funnelling effect* », où les voisins du sink représentent le goulot d'étranglement du trafic. Les entités mobiles peuvent réduire *the funnelling effect*, elles peuvent visiter des régions différentes dans le réseau et répartissent la consommation énergétique d'une manière uniforme.

1.4. Les data ferries

Les réseaux de capteurs sont tributaires à la fois de la faible portée des liens radio et des conditions d'accès parfois difficiles. Une solution consiste à utiliser une ou plusieurs entités mobiles appelées « data ferries » et chargées de la collecte des données issues des capteurs.

Ces data ferries étant par nature des puits pour ces données, ils jouent le rôle des nœuds relais dont la fonction est la collecte de données à partir des nœuds capteurs par une transmission à un seul saut, le stockage de ces données et le transport des données collectées vers la station de base. La transmission à un seul saut conduit à des économies d'énergie substantielles au niveau des capteurs. Le type de la mobilité attribué aux data ferries permet généralement à ces derniers d'interconnecter des parties physiquement éloignées d'un même réseau.

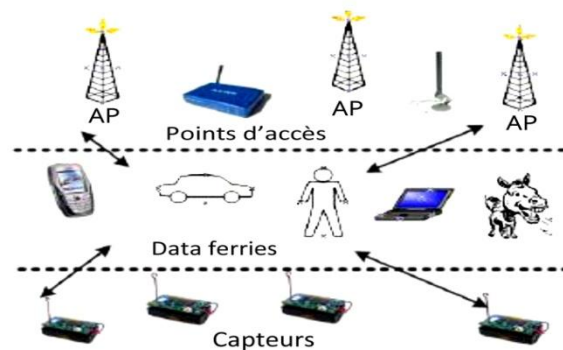


Figure 1.8. Architecture trois tiers des data ferries [SHA03].

Le principal désavantage de cette approche (figure 1.8) est l'augmentation de latence parce que les capteurs doivent attendre que les data ferries soient dans leur champ de communication avant de commencer le transfert des données. Néanmoins, pour plusieurs applications de la collecte de données (qui nécessitent des données à analyser pour une période qui s'étale à des heures ou même une journée) cette augmentation de latence est tolérée. Donc, l'utilisation des data ferries est destinée à des applications tolérantes aux délais « delay tolerant » où la consommation d'énergie des capteurs est la contrainte primordiale [SHA03].

1.4.1. Architecture à trois niveaux des data ferries

L'architecture des data ferries fournit une connectivité d'une large envergure pour les réseaux de capteurs par l'exploitation des agents mobiles tels que : personnes, animaux, robots ou véhicules en déplacement dans l'environnement. Ces nœuds mobiles agissent comme une couche intermédiaire pour former un système de transport des paquets pour connecter le réseau. Ceci crée une abstraction en couche à trois niveaux, qui peut être ajustée pour les différents types de situation :

- La couche supérieure : les points d'accès ;
- La couche intermédiaire : les agents mobiles (data ferries);
- La couche inférieure : les nœuds capteurs sans fil.

La couche supérieure est constituée d'un ensemble de stations de base qui sont connectées à l'utilisateur du réseau via internet ou un satellite. Les données sont récoltées dans les stations de base par le biais des data ferries pour être exploitées par les utilisateurs finaux. Ces points d'accès peuvent communiquer avec un entrepôt de données central qui permet de synchroniser les données qu'ils collectent, détecte les duplicates et retourne des acquittements aux data ferries (un acquittement peut être nécessaire pour assurer la fiabilité des données pour certaines applications) [SHA03, ANA07].

La couche intermédiaire est composée d'agents mobiles (data ferry) dont la fonction est la collecte des données et le transport des données collectées vers les points d'accès. Les data ferries ont une large capacité de stockage (relativement aux capteurs), une énergie renouvelable et la possibilité de communiquer soit avec les capteurs ou les points d'accès. Un réseau entre les data ferries peut être établi pour l'échange des données entre eux par des chemins multi-saut. Ceci améliore les performances du système par la réduction du temps de latence et l'augmentation de la fiabilité.

La couche inférieure consiste en une distribution aléatoire des nœuds capteurs. Les capteurs sont restreints en termes d'énergie et de capacité mémoire. Le travail réalisé par cette couche doit être minimisé vu les contraintes en ressources par rapport aux autres couches.

L'avantage de l'architecture trois tiers des data ferries est sa robustesse et le passage à l'échelle ("scalability") comparée à une solution centralisée. Aucun des capteurs ne dépend des data ferries, d'où la défaillance d'un data ferry ne déconnecte pas le capteur du réseau. Aussi, l'architecture passe aisément à l'échelle (scalable), l'ajout de nouveaux nœuds capteurs ou data ferry ne nécessite aucune configuration réseau. Les points d'accès et les data ferry peuvent aisément être déployés pour accroître le débit et réduire la latence.

Avantages de cette architecture [SHA03] :

- Bien que l'approche de la collecte de données par les data ferries subisse une latence supérieure, cette approche offre une consommation d'énergie optimisée pour les capteurs et un coût d'infrastructure réduit. Ces caractéristiques sont importantes pour plusieurs applications.
- Etant donné une densité suffisante des data ferries, le système est plus robuste qu'un réseau de capteurs traditionnel. Comme les capteurs sont reliés uniquement aux data ferries et les data ferries sont interchangeable, la défaillance de n'importe quel data ferry ne signifie pas une perte de la connectivité pour le capteur. La défaillance augmente la latence et réduit le taux d'accès aux données pour le réseau.
- La flexibilité du système permet à la couche intermédiaire des data ferries d'être utilisée simultanément par différentes applications. La couche intermédiaire peut être vue comme un mécanisme de transport mobile pour connecter des nœuds hétérogènes.
- Le transfert des données a lieu sur des distances courtes pour à une économie d'énergie.

1.4.2. La collecte de données par les data ferries

La collecte de données est assurée par les data ferries, le processus de la collecte de données peut être divisé en deux phases principales : la découverte (discovery) et le transfert des données (data transfert). La collecte de données commence quand le data ferry est dans le champ de communication du nœud capteur. La durée pendant laquelle les nœuds sont dans la même portée de connexion est appelée le temps de contact [MAR11, ANA08].

Du fait que les nœuds ne peuvent communiquer que s'ils sont dans la même portée de communication, la découverte est définie comme étant le processus qui permet à un nœud de détecter un contact, et le transfert des données comme étant l'échange des messages entre les nœuds en contact. Le temps alloué au transfert des données est appelé temps résiduel de contact.

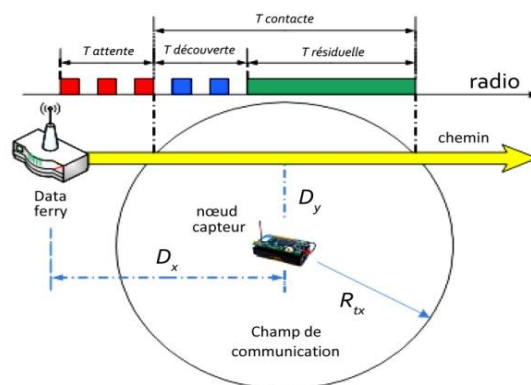


Figure 1.9. Scénario du processus de la collecte de données [MAR11].

- a) **La découverte** : c'est la première phase dans le processus de la collecte de données, vu que la présence d'un data ferry dans le champ de communication est généralement inconnue par les capteurs. Le but du processus de la découverte est la détection des contacts (voisins) dès qu'ils se produisent et avec une faible dépense énergétique.
- b) **Le transfert des données** : c'est la deuxième phase dans le processus de la collecte de données. Cette phase suit immédiatement le processus de découverte. Le but du processus de transfert des données est de prolonger la durée de temps résiduel de contact, afin de maximiser le débit (les messages transférés avec succès pour chaque voisin en communication) avec optimisation de la consommation énergétique.

1.4.3. Modes pour la collecte des données

Il y a deux modes de collecte de données. Selon que les données soient transmises directement au data ferry ou bien par le biais d'autres capteurs. Dans le premier mode, le data ferry se déplace à travers tout le réseau pour la collecte de données. Alors que, dans le deuxième mode, le data ferry visite uniquement des capteurs bien déterminés appelés les points de rendez-vous pour la collecte de données [MUN07] :

- a) **La collecte de données à un saut** : Dans ce mode, chaque nœud capteur est isolé. Un nœud capteur ne communique pas avec ses nœuds voisins. Les capteurs envoient directement leurs données aux data ferries par une transmission à un seul saut. Ce mode fournit une consommation d'énergie optimisée. Mais le plus grand inconvénient de ce mode de collecte de données se situe dans l'augmentation de la latence. L'illustration est montrée dans la figure 1.10.

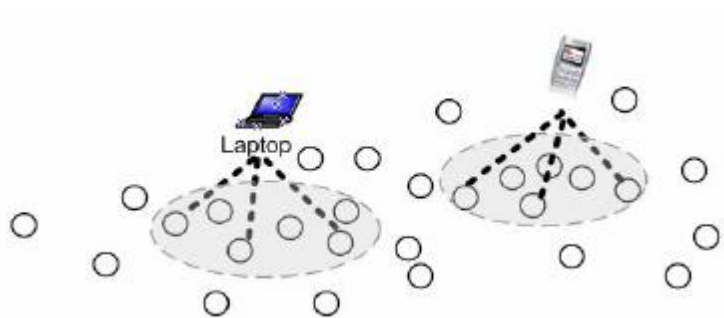


Figure 1.10. La collecte de données à un seul saut.

- b) **La collecte de données multi-saut** : Ce mode permet des communications multi-saut entre les nœuds capteurs. Les nœuds capteurs peuvent s'organiser en réseau ad hoc à la phase initiale. Les capteurs envoient leurs données avec des communications multi-saut aux data ferries. Dans ce mode, un bas niveau de latence est assuré mais le maintien et la mise-à-jour de la topologie de réseau consomme beaucoup d'énergie. L'illustration est montrée dans la figure 1.11.



Figure 1.11. La collecte de données à multi-saut.

1.4.4. Exemples d'application des data ferries

Les véhicules autonomes sont un type particulier des data ferries, nous citons dans ce qui suit, quelques applications de l'utilisation des véhicules aérien (drone) et des véhicules sous-marins autonomes dans l'approche de la collecte de données [FIN10].

1.4.4.1. Les drones ou UAVs (*Unmanned Aerial Vehicles*) :

Les UAVs sont des engins volants capables d'effectuer des missions de manière autonome sans présence humaine à bord. Ces plateformes volantes possèdent de réels avantages par rapport à d'autres systèmes aériens et ouvrent de nombreuses perspectives. Le nombre d'industriels et d'universités qui s'y intéressent ne cesse d'augmenter à cause des divers domaines d'applications, tel que : la poursuite vidéo d'objet en mouvement de manière autonome, l'exploration de l'environnement, la gestion des risques naturels, la surveillance du trafic routier, les relais de communications...etc. Les applications des UAVs dans le domaine militaire sont aussi nombreuses telles que : Exploration des zones sensibles et zones à accès difficiles et dangereuses (zones contaminées et zones minées), surveillance des frontières et neutralisation des activités terroristes [DER11, ERM08].



Figure 1.12. Exemple sur les drones.

L'exemple dans la figure 1.13 présente une application pratique de l'utilisation des drones dans la collecte des données. L'application consiste en l'utilisation d'un réseau de capteurs pour la surveillance de l'état des infrastructures critiques, à l'exemple d'un pont. Les nœuds capteurs sont déployés le long du pont. Un drone est utilisé pour la collecte des données issues des capteurs pour l'évaluation de la structure après qu'un événement telle une catastrophe naturelle ait lieu. Les données sont transportées vers la station de base pour mesurer la fiabilité de l'infrastructure civile en question. Dans ce type d'application, les capteurs sont dotés d'un système de localisation GPRS et les drones sont programmés pour atteindre la position de chaque capteur [TOD06].

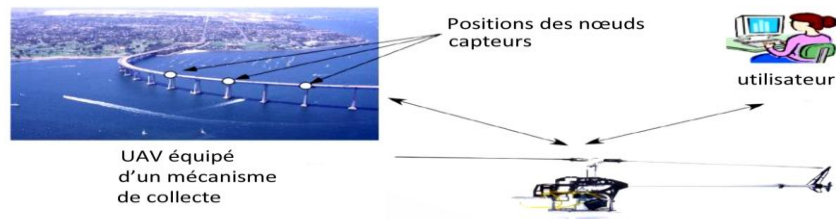


Figure 1.13. La surveillance d'un pont par les data ferries.

1.4.4.2. Les véhicules sous-marins autonomes ou AUVs(*Autonomous Underwater Vehicles*) :

Les AUVs sont des engins autopropulsés, équipés d'une source d'énergie embarquée. Cette catégorie d'engins est exploitée dans l'investigation du milieu sous-marin qui est une préoccupation de nombreux domaines comme la recherche océanographique, les applications militaires. L'utilisation de véhicules télé-opérés dans l'automatisation de tâches d'inspection, de reconnaissance, de détection ou encore de mesure de paramètres physico-chimiques est fortement justifiée dans ce milieu immense [VAS05, MAT06].



Figure 1.14. Exemple d'un véhicule sous-marin autonome.

L'exemple dans la figure 1.15 montre l'utilisation des véhicules sous-marins autonomes dans la collecte des données à partir d'un réseau de capteurs sous-marin, le véhicule sous-marin joue le rôle d'un data ferry où il collecte les données à partir des nœuds capteurs sous-marins. Les données collectées sont ensuite transportées vers la station de base : un bateau pour l'analyse et l'exploitation. Ce type d'application peut être utilisé dans la formation d'images sismique de gisement de pétrole sous-marin [HEI05, BOR08, CRE03].

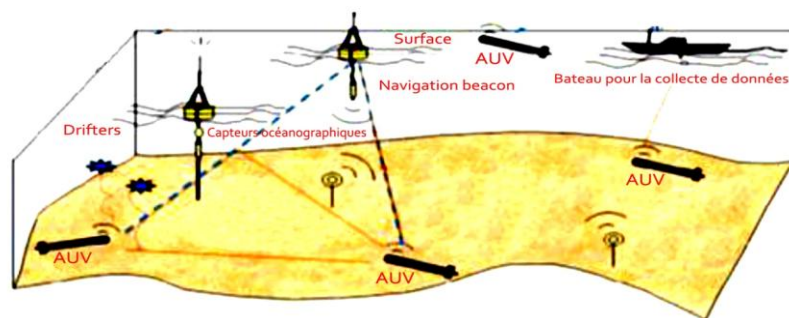


Figure 1.15. La surveillance sous-marin par les data ferries [HEI05].

1.5. Conclusion

Dans ce chapitre nous avons présenté des généralités sur les réseaux de capteurs mobiles et le processus de la collecte de données dans les RCSFs. Nous avons commencé par donner des généralités sur les réseaux de capteurs mobiles avec spécification de l'avantage d'utilisation de la mobilité dans les réseaux de capteurs. Ensuite, nous avons mentionné les différents types et architectures de mobilité existants. Enfin, nous avons étudié le processus de la collecte de données dans les réseaux de capteurs mettant l'accent sur la collecte de données par les data ferries où nous avons recensé les bénéfices de cette architecture de collecte de données.

Chapitre 2 La planification de la trajectoire des data ferries

2.1. Introduction

Les réseaux de capteurs sans fil ont été proposés comme moyen puissant dans les situations d'observation des événements et des environnements sur de longues périodes de temps [AKY02]. Parmi les nombreux défis rencontrés lors de la conception de réseau de capteurs et de protocoles, le maintien de la connectivité et l'optimisation de la durée de vie du réseau apparaissent comme des considérations critiques [EKI06]. L'utilisation des data ferries peuvent être mises à profit pour surmonter ces deux problèmes, la connectivité et l'optimisation énergétique peuvent être adressée comme suit :

- **La connectivité** : les data ferries peuvent être utilisés pour transporter des données entre des parties isolées de réseau de capteurs.
- **Optimisation énergétique** : les données transportées dans les data ferries peuvent réduire la consommation énergétique des nœuds capteurs par la réduction des communications multi-saut.

De toute évidence, le modèle de mobilité des data ferries a un impact significatif sur le schéma optimal de la collecte de données. Dans la littérature, il existe trois types de mobilité des data ferries : aléatoire, prédictive et contrôlée. Lorsque la mobilité est contrôlable, les mouvements du data ferry peuvent être conçus de manière à atteindre des objectifs précis et d'optimiser les paramètres de performance donc la mobilité contrôlée des data ferries donne plus de flexibilité pour la conception d'un schéma de collecte de données.

Dans ce chapitre, nous présentons un état de l'art sur les approches de la planification de la trajectoire des data ferries. Nous commençons par faire une taxonomie sur les approches de la planification. Ensuite, nous examinons l'approche de la planification basée TSP où nous proposons une classification et nous effectuons une étude et synthèse des travaux existants.

2.2. Types de la mobilité des data ferries

Les réseaux de capteurs sont utilisés pour collecter des données à des points distants sur une grande surface. L'observation de la densité du trafic dans une grande ville en est un exemple. Les capteurs sont placés sur les routes pour observer les véhicules. Ces scénarios de déploiement étendus souffrent de problèmes de connectivité. Une solution possible est d'utiliser les data ferries qui collectent les données à partir des capteurs en les visitant individuellement [EKI06]. Différents types de mobilité ont été pris en compte dans le domaine des réseaux sans fil pour faciliter le transfert de données. Ceux-ci peuvent être classés en trois catégories : *aléatoire*, *prédictible* et *contrôlée* [SOM06].

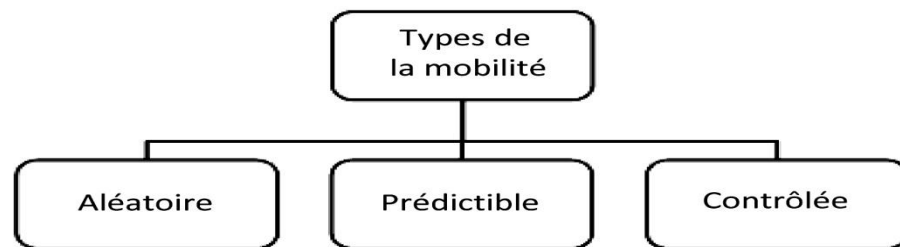


Figure 2.1. Types de la mobilité des data ferries.

- a) **La mobilité aléatoire** : est caractérisée par des contacts qui n'ont pas lieu régulièrement, mais avec une distribution probabiliste [MAR11]. Les data ferries se déplacent aléatoirement et collectent les données à partir des nœuds capteurs qui sont dans leur champ de communication. Les données collectées sont ensuite transportées vers le point d'accès [EKI06].
- b) **La mobilité prédictible** : est caractérisée par la régularité dans les contacts de l'entité mobile, qui entre dans le champ de communication des nœuds capteurs à des temps spécifiques (généralement périodique) [MAR11]. Dans l'exemple de l'observation de la densité du trafic, les données sont collectées à partir des capteurs par les véhicules qui passent près de capteurs. La trajectoire des data ferries (véhicules) est connue par les capteurs, donc le moment de transfert des données peut être prédit [EKI06].
- c) **La mobilité contrôlée** : c'est le cas où le mouvement de data ferry est contrôlé et programmé pour réaliser certaines tâches dans le réseau. Dans ce contexte, différents problèmes se posent, principalement liés à la façon de planifier le mouvement des data ferries (trajectoire et vitesse) pour satisfaire certaines contraintes de qualité de service (comme la réduction de latence et le débordement de tampon) avec maintien de la consommation énergétique la plus faible possible [MAR11].

2.3. Classification des approches de la planification de la trajectoire des data ferries

Les approches de la collecte de données par les data ferries peuvent être classées en : (i) La collecte de données par communication directe, (ii) La collecte de données basée points de rendez-vous. Dans la collecte de données par communication directe, le data ferry visite tous les capteurs et collecte les données directement à partir de ces capteurs. Cette approche élimine les communications multi-sauts, et optimise ainsi la consommation d'énergie. Cependant, elle génère une latence supérieure pour les déplacements lents de data ferry. Dans la collecte de données basée points de rendez-vous, le data ferry visite des points spécifiques de collecte de données appelés les points de rendez-vous. Les points de rendez-vous permettent au data ferry de collecter un volume important de données sans traverser une longue distance, diminuant ainsi considérablement la latence de la collecte de données.

Chaque classe de la collecte de données est encore divisée en trois sous-classes selon les techniques utilisées (figure 2.2).

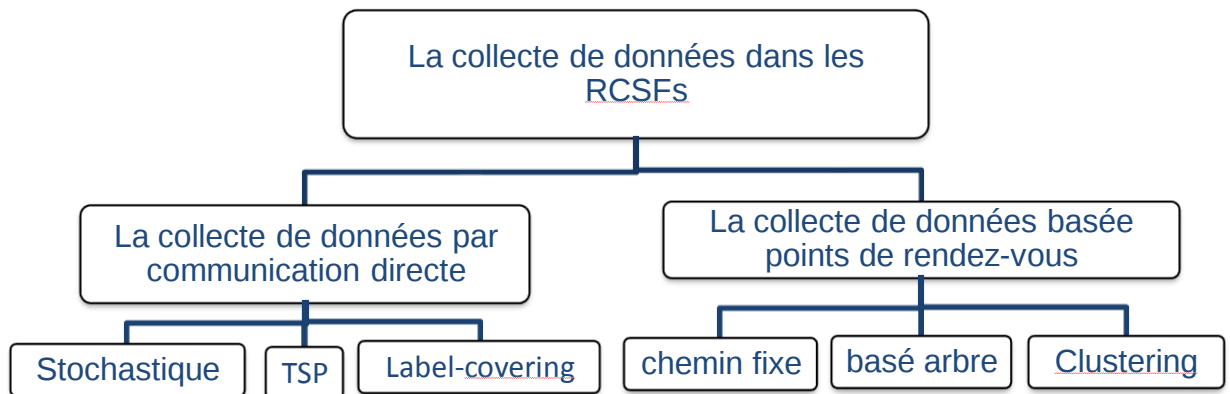


Figure 2.2. Taxonomie des approches de la planification de la trajectoire des data ferries.

2.3.1. La collecte de données par communication directe (Direct-Contact Data Collection)

Dans la collecte de données par communication directe, le data ferry collecte les données directement à partir des nœuds capteurs par une communication à un seul saut. Les ferries peuvent retransmettre ou, si nécessaires, transporter physiquement les données à une station de base fixe. Cette approche réduit au minimum la consommation d'énergie au niveau des capteurs, puisque les capteurs n'ont pas besoin de transmettre des messages entre eux. Dans ce scénario, la principale préoccupation est le calcul de la meilleure

trajectoire de ferry qui couvre tous les points de collecte et réduit au minimum la latence de la collecte des données.

2.3.1.1. Trajectoire stochastique de collecte de données (Stochastic Data Collection Trajectory)

Les auteurs dans [SHA03] ont considéré la mobilité stochastique de ferry et ont proposé un algorithme de collecte de données simple. Dans leur proposition, les capteurs stockent leurs données localement et attendent l'arrivée du data ferry. Le ferry se déplace aléatoirement et collecte les données des capteurs dans son champ de communication. Les données collectées sont ensuite transportées par le data ferry à un point d'accès.

Dans le cas de la mobilité stochastique de ferry, la consommation d'énergie de côté des capteurs est seulement due à la découverte de ferry et au transfert de données. Le ferry envoie des messages de «beacon» tout en se déplaçant. Une manière simple de découvrir le data ferry est de surveiller la voie de transmission sans fil. A chaque fois qu'un capteur reçoit un message de «beacon», il conclut que le ferry arrive. Cependant, la surveillance constante de canal est très coûteuse en énergie. Le travail dans [CHA03] montre que, si le data ferry (par exemple, monté sur un autobus) se déplace le long d'un chemin régulier, les capteurs peuvent prévoir leur arrivée.

2.3.1.2. Chemin TSP pour la collecte de données (TSP tour for data collection)

Quand la mobilité de data ferry est contrôlable, la latence de la collecte de données peut être réduite par la sélection du meilleur chemin de data ferry. Il n'est pas difficile de conclure que la collecte de données par communication directe est généralement équivalente au problème de voyageur de commerce TSP (Traveling Salesman Problem). Le problème TSP étant : un certain nombre de villes (capteurs), trouver le chemin le plus court qui visite chaque ville (capteur) exactement une fois et revient à la ville initiale. Les auteurs de [NES07] ont formulé le problème de la planification de la trajectoire de data ferry comme variante de TSP, connu en tant que TSPN (Traveling Salesman with Neighborhood), où le data ferry doit visiter le voisinage de chaque capteur exactement une fois. Il suffit que le data ferry soit dans le champ de communication (modélisé comme disque) d'un capteur pour pouvoir collecter les données à partir de ce capteur. Figure 2.3 montre la différence entre les chemins TSP (chemin en pointillé) et TSPN d'un data ferry.

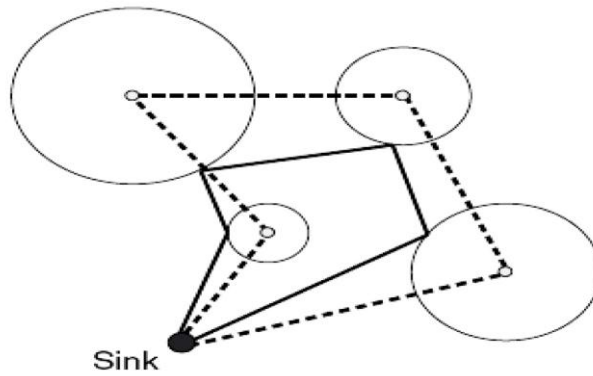


Figure 2.3. La différence entre les chemins TSP et TSPN.

2.3.1.3. Chemin Label-Covering pour la collecte de données (Label-Covering tour for Data collection)

Les auteurs dans [SUG07] et [SUG08] ont adressé la planification du chemin de data ferry avec minimisation de la latence. Ils ont écarté la condition que le ferry visite une fois le voisinage de chaque capteur. L'intuition est que le temps de déplacement de ferry pourrait être long si la longueur de l'intersection de son chemin avec le champ de communication de capteur est courte, car dans ce cas, le ferry doit ralentir pour la collecte des données. Visiter une fois le voisinage de chaque capteur ne peut pas être toujours une bonne stratégie. Une visite multiple avec un contrôle approprié de la vitesse peut rapporter une meilleure solution.

Les auteurs ont simplifié le problème de la sélection du chemin en réduisant l'espace de recherche à un graphe géographique complet, où il y a des sommets à la position des capteurs et du sink. Le data ferry se déplace le long des arêtes d'un sommet à l'autre. A Chaque arête est associée un coût et à un ensemble d'étiquettes. Le coût est défini comme étant la distance euclidienne de l'arête ; l'ensemble d'étiquette représente l'ensemble des nœuds capteurs ayant leurs champs de communication en intersection avec l'arête. Donc lors de déplacement le long d'une arête, le data ferry collecte les données à partir des capteurs dans leurs étiquettes qui sont associées à cette arête.

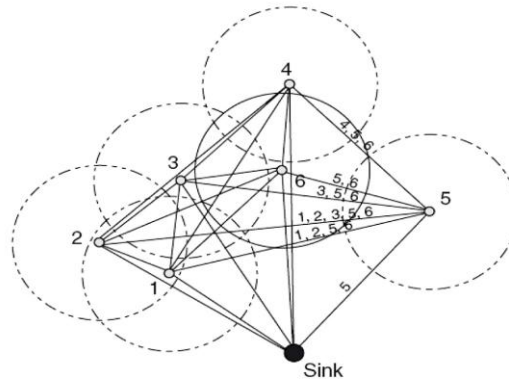


Figure 2.4. Le chemin Label-Covering [SUG07].

L'objectif est de trouver le chemin avec le coût minimum le long duquel le ferry peut collecter les données de tous les nœuds. En d'autres termes, le chemin le plus court dont l'ensemble d'étiquette associé couvre tous les capteurs. Dans ce contexte, le ferry ne visite pas nécessairement tous les sommets. Les auteurs ont montré que ce problème est NP-complet, et ont proposé un algorithme approximatif pour le résoudre.

2.3.2. La collecte de données basée points de rendez-vous (Rendezvous-Based Data Collection)

La collecte de données par communication directe a un grand avantage pour les économies d'énergie. Cependant, elle augmente de manière significative la latence de collecte de données à cause de la vitesse lente de déplacement du data ferry. La collecte de données basée points de rendez-vous est proposée pour faire un compromis entre la consommation d'énergie et la latence. Les capteurs envoient leurs données à un sous ensemble des capteurs appelés les points de rendez-vous (RPs: Rendez-vous points) ; le data ferry se déplace dans le réseau et collecte les données à partir des points de rendez-vous rencontrés. L'utilisation des points de rendez-vous permet au data ferry de collecter un volume important de données sans traverser une longue distance, diminuant ainsi considérablement la latence de la collecte de données. Les travaux de recherche se concentrent principalement sur la sélection des points de rendez-vous.

2.3.2.1. La sélection des RPs par chemin fixe

Les auteurs dans [KAN04] ont proposé d'utiliser un chemin en ligne droite de data ferry pour la collecte des données. A la phase d'initialisation, le data ferry diffuse un message «beacon» tout en se déplaçant le long de la ligne droite (figure 2.5). Le message a un compteur de saut qui indique le nombre de sauts que le message a traversé. Chaque

nœud récepteur incrémente le compteur de saut avant rediffusion. Il se rappelle également du nœud dont il reçoit le message. Après la phase d'initialisation, un certain nombre d'arbres sont construits autour de la trajectoire de data ferry.

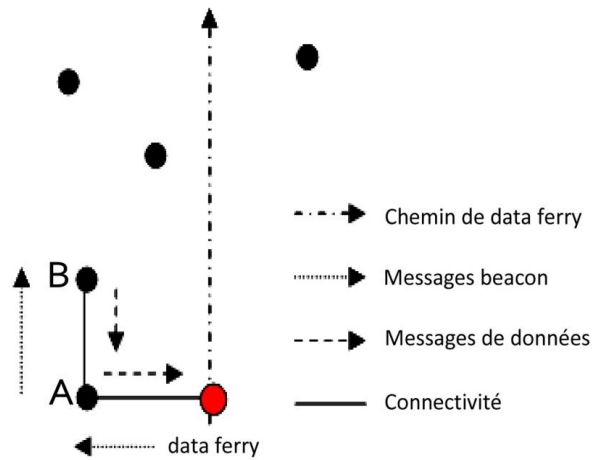


Figure 2.5. La sélection des RPs par chemin fixe.

La racine de chaque arbre est considérée comme RP. Les capteurs envoient leurs données le long du chemin ascendant à la racine de leur arbre. Pendant que le data ferry se déplace, les RPs envoient leurs propres données ainsi que les données reçues de leurs membres au data ferry. Deux algorithmes sont présentés pour ajuster la vitesse du data ferry afin d'augmenter la quantité des données collectées. Dans l'algorithme SCD («Stop to Collect Data»), le principe est que le data ferry se déplace à une vitesse constante et s'arrête aux capteurs ayant des données à collecter. Dans l'algorithme ASD («Adaptative Speed Control»), le principe est de classifier les nœuds en trois groupes en fonction de pourcentage des données dans la mémoire tampon et de définir une vitesse différente pour chaque groupe.

2.3.2.2. La sélection des RPs basée arbre

Les auteurs dans [XING07] ont étudié la sélection des RPs basée sur la structure d'arbre avec contrainte sur la latence définie par D. Un arbre géométrique dont la racine est le sink et composé de tous les capteurs de réseau est défini. N'importe quel point dans l'arête de l'arbre géométrique peut servir comme RP. Un algorithme est présenté pour la planification de chemin de data ferry pour visiter un sous-ensemble des RPs, telle que la longueur de chemin soit inférieure à celle de L_{max} . Il fonctionne en itérations. Dans chaque itération, un nœud dans l'arbre avec la plus grande utilité est inclus dans une liste de RP (qui, contient initialement seulement le sink). L'utilité d'un RP est définie comme rapport de l'énergie en l'ajoutant au chemin de ferry à l'augmentation de longueur

de chemin. La longueur de chemin des RPs est calculée par l'utilisation de n'importe quelle solution TSP. Tous les RPs dont les utilités deviennent zéro sont immédiatement enlevés de la liste. Si la longueur maximale de chemin est atteinte, ou bien tous les nœuds sont inclus dans la liste, l'algorithme est terminé ; sinon, une nouvelle itération est commencée pour trouver plus de RPs. Par l'ajustement de la valeur de L_{max} , un compromis souhaitable est réalisé entre la qualité de la solution et la complexité de calcul.

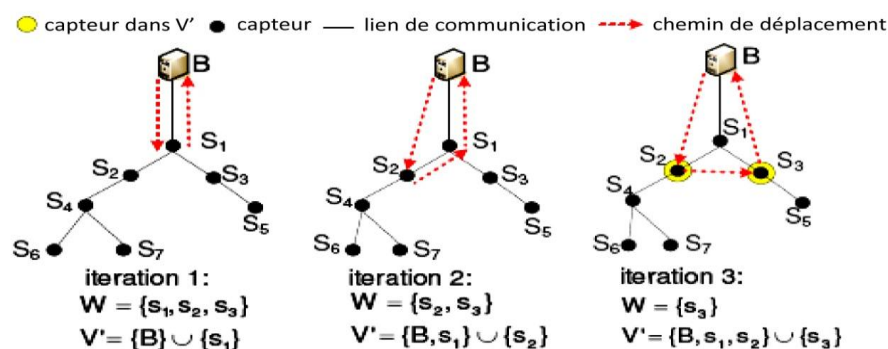


Figure 2.6. La sélection des RPs basé arbre [XING07].

L'auteur de [XING08] considère que les capteurs peuvent agréger leurs données avant de les envoyer au data ferry. Il explique comment planifier le chemin du ferry de façon à ce que la longueur du chemin soit inférieure à L_{max} et la quantité totale des données envoyées au ferry puisse être minimisée. Ce travail propose la planification du chemin de data ferry par adoption de la structure Steiner tree pour agréger les données.

2.3.2.3. La sélection des RPs par Clustering

Les auteurs dans [RAO08] ont présenté une approche générique pour la collecte de données par les data ferries. L'approche propose un protocole de navigation distribué pour naviguer le data ferry à travers des capteurs de navigation appelés les agents de navigation NAs (Navigation Agents). Le protocole est basé sur le schéma k-hop data collection pour la collecte des données où les capteurs qui se trouvent dans k sauts (k-hops) d'un NA peuvent envoyer leurs données au NA (par conséquent le data ferry). En fonction des valeurs du paramètre k, l'approche migre d'un schéma de collecte de données par communication directe, vers un schéma de collecte basé points de rendez-vous.

Le data ferry navigue entre ces NAs par des capteurs intermédiaires INs (Intermediate Navigators), un protocole de trois étapes a été proposé.

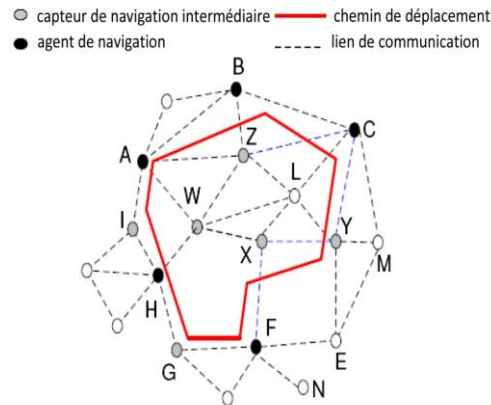


Figure 2.7. La sélection des RPs par Clustering [RAO08].

2.4. La planification de la trajectoire basée TSP

Les approches de la planification de la trajectoire basées TSP sont destinées aux applications de surveillance constituées d'un ensemble de nœuds statiques localisés à des positions bien déterminées, et une entité mobile (drone, véhicule) utilisée pour la collecte des données. Dans la littérature, il existe deux classes de travaux : (i) La classe TSP : travaux qui ne considèrent pas les communications, modélisent les capteurs par des points ponctuels. (ii) La classe TSPN : travaux qui considèrent les spécificités des communications. Ces derniers modélisent la portée de communication des capteurs par une forme géométrique (disque). Dans cette section nous allons étudier les deux classes avec la présentation des principaux travaux existants dans la littérature.

2.4.1. La classe des travaux TSP

Les travaux dans cette catégorie visent à planifier la trajectoire d'une *data ferry* avec optimisation de certains paramètres de réseau tel que le temps de latence. Des stratégies sont utilisées pour optimiser ces paramètres, par la suite, un algorithme TSP est appliqué pour la planification. Selon la stratégie adoptée pour la planification de la trajectoire et les paramètres à optimiser, nous avons proposé une nouvelle classification pour cette classe de travaux. La planification de la trajectoire basée TSP peut être regroupée en trois catégories de planification : ordonnancement, adaptative et Single-hop.

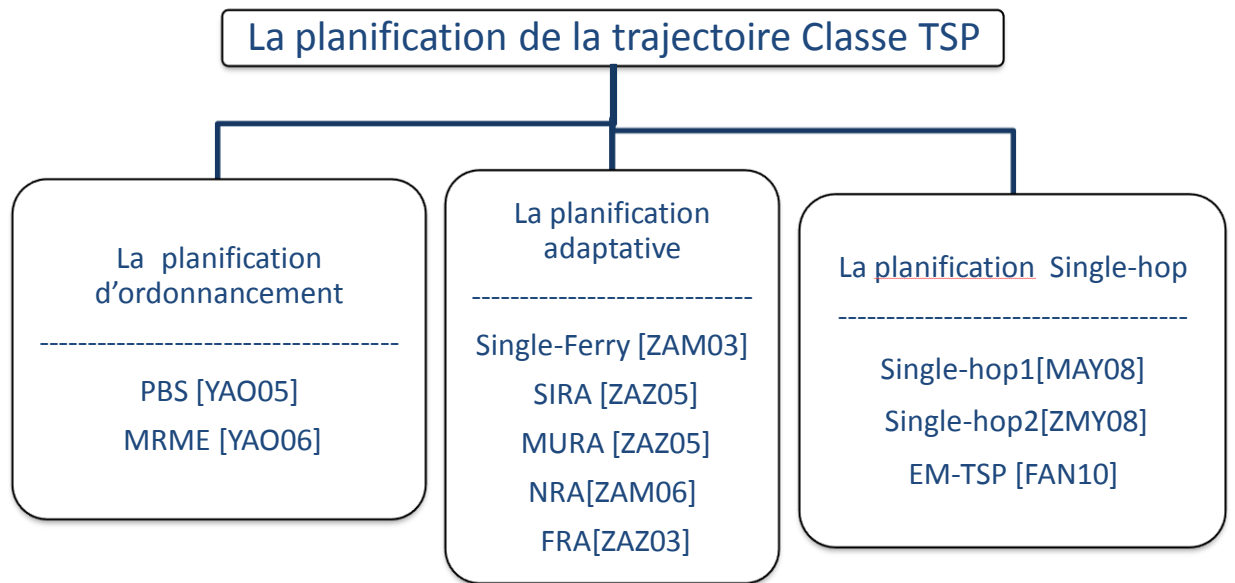


Figure 2.8. Nouvelle classification des approches de la collecte de données de la classe TSP.

Dans cette partie, nous allons détailler les travaux pour chaque catégorie. La planification d'ordonnancement procède à un partitionnement de réseau en classes de nœuds, ensuite une stratégie d'ordonnancement des visites de data ferry est définie pour optimiser certains paramètres. La planification adaptative vise à planifier le chemin de data ferry en l'adaptant à une solution initiale, la trajectoire finale est améliorée par l'application des techniques d'optimisation. La planification single-hop assure la planification de chemin de data ferry de telle sorte que les communications entre le data ferry et les capteurs soient des communications à un seul saut.

2.4.1.1. La planification d'ordonnancement

La planification d'ordonnancement vise à planifier les visites du data ferry aux nœuds de réseau avec regroupement des nœuds en classes en fonction des paramètres à optimiser, et de définir ensuite une stratégie d'ordonnancement pour visiter chaque classe de nœuds. Une heuristique TSP est appliquée sur chaque classe de nœuds pour la planification et des algorithmes de partitionnement sont utilisés pour la classification.

a) L'algorithme PBS

L'algorithme PBS «Partitioning-Based Scheduling» vise à planifier les visites de data ferry pour chaque capteur afin d'éviter les pertes de données dues au débordement de la mémoire tampon. L'idée de base de l'algorithme PBS est de partitionner d'abord le réseau en plusieurs groupes avec une contrainte sur l'échéance de débordement de tampon définie dans chaque groupe. Par la suite, une trajectoire est définie au sein de chaque groupe, pour en déduire à la fin, la trajectoire finale de data ferry qui est la concaténation des différentes trajectoires.

Pour respecter la contrainte sur le débordement de tampon, les groupes seront visités à des fréquences différentes, où les groupes ayant une échéance de débordement de tampon faible seront visités plus que les groupes avec une échéance supérieure.

Comme c'est présenté dans [YAO05], L'algorithme PBS aborde le problème en le divisant en deux sous-problèmes : Partitionnement et Planification. Dans cet algorithme, les nœuds sont partitionnés en plusieurs groupes, appelés bins, tel que les nœuds dans le même groupe (bin) ont des échéances de débordement du tampon similaires et sont géographiquement proches les uns des autres. Ensuite, un algorithme TSP est appliqué sur chaque bin afin de planifier la trajectoire de ferry au sein de chaque bin. Finalement, les trajectoires de ferry dans les différents groupes sont concaténées pour former la trajectoire complète.

Les étapes de l'algorithme PBS sont les suivantes :

- Partitionnement des nœuds en M bins en fonction de l'échéance de débordement de tampon.
- Partitionnement géographique de chaque bin en sous bins: par l'utilisation d'un algorithme 2D-tree où les deux dimensions sont la longueur et la largeur de la scène.
- Calcule du chemin TSP pour chaque sous bin.
- Concaténation des chemins des sous bins pour produire le chemin final.

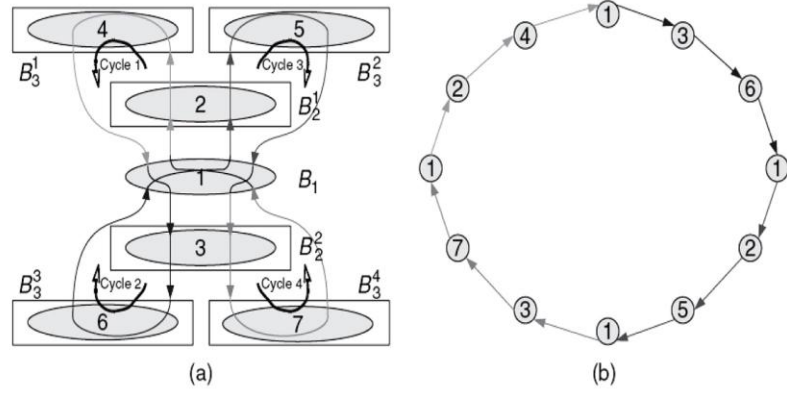


Figure 2.9. La planification de la trajectoire par l'algorithme PBS.

Le data ferry commence à partir de capteur avec le temps minimum de débordement de tampon dans un sous bin de B_1 . Il voyage le long d'un super cycle composé d'un certain nombre de cycles de visite des bins. Chaque cycle de visite contient exactement un sous bin de chaque bin B_i dans l'ordre. Dans chaque cycle de visite, un sous bin en B_i est suivi du sous bin géographiquement le plus proche dans B_{i+1} . Puisqu'il y a deux fois plus de sous bin dans B_{i+1} qu'en B_i , chaque sous bin dans B_i est suivi exactement par deux sous bin de B_{i+1} dans le super cycle.

Le problème de déplacement du data ferry est réduit au problème TSP dans chaque sous bin. Une fois le chemin est construit, la vitesse minimum pour la collecte de données sans perte peut être déterminée par L_{max}/o_{min} , où L_{max} est la longueur du plus long chemin entre deux visites consécutives à un capteur, et o_{min} est le temps minimum de débordement de tampon.

b) L'algorithme MRME

L'algorithme MRME «Mutli-hop Route to Mobile Element» présenté dans [YAO06] est une variante de l'algorithme PBS. L'algorithme MRME considère le schéma hybride de communication où des communications multi-sauts entre les nœuds capteurs peuvent avoir lieu, dont le but de transporter des messages urgents vers le data ferry. L'idée de cet algorithme est de sélectionner un ensemble de nœuds dans le réseau pour transporter les messages urgents. La sélection de ces nœuds est effectuée par la réduction de leurs échéances de débordement de tampon et comme dans l'algorithme PBS, les nœuds avec échéance faible se sont les plus visités par le data ferry. Ceci assure que les messages urgents arrivent au data ferry par cet ensemble de nœuds.

L'exécution de l'algorithme MRME s'effectue en trois phases. Dans la première phase, les capteurs sont partitionnés en sous bins; dans la deuxième phase l'échéance de débordement de tampon des capteurs jouant le rôle de relais pour le transport des messages urgents est itérativement réduit ; dans la troisième phase, l'algorithme PBS est appliqué sur les valeurs modifiées de l'échéance de débordement de tampon pour planifier la trajectoire de ferry.

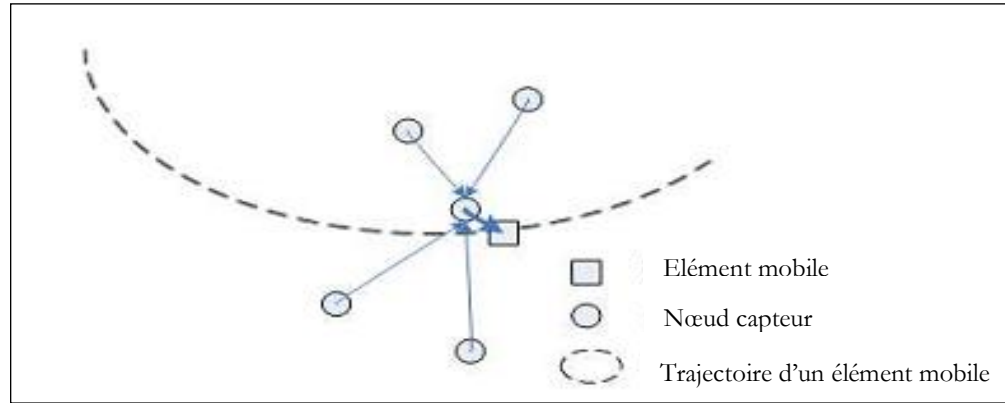


Figure 2.10. La transmission multi-saut pour le transport des messages urgents.

MRME est un algorithme offline, donc les nœuds ne sont pas impliqués dans le calcul. La seule information exigée par un nœud est l'identité d'un autre nœud pour l'expédition des messages urgents.

2.4.1.2. La planification adaptative

La planification adaptative considère que tous les capteurs sont isolés et le chemin final est ajusté à partir de la solution initiale. Un algorithme TSP est utilisé pour la planification de chemin initiale. Des techniques d'optimisation sont appliquées sur le chemin planifié par TSP afin d'adapter la trajectoire finale en fonction des paramètres à optimiser. Les solutions de cette famille considèrent le problème de la planification de la trajectoire d'un seul data ferry et de plusieurs data ferries.

a) La planification avec un seul data ferry

Etant donnée un data ferry F et un ensemble de capteurs isolés $S = \{s_1, s_2, \dots, s_n\}$, le travail effectué dans [ZAM03] propose comment prévoir un chemin pour que F visite S . Le but est de collecter les données des capteurs dans un délai moyen minimisé et avec un temps de transmission de données des capteurs vers le ferry optimal. Ce problème est décomposé en deux sous-problèmes d'optimisation : un problème d'optimisation de délai moyen et un autre problème d'optimisation de temps de transmission :

- 1) Optimisation de délai moyen : Pour n'importe quel chemin p calculé par un algorithme approximatif de TSP. Le délai de message entre deux capteurs s_i et s_j est défini par $T_{ij}^p = \frac{|p|}{2v} + \frac{d_{ij}^p}{v}$ où $|p|$ est la longueur totale de p , v représente la vitesse de F , et d_{ij}^p est la distance entre s_i et s_j sur le chemin p . Ici, $\frac{|p|}{2v}$ est le délai moyen d'attente pour que s_i soit visité par F , et $\frac{d_{ij}^p}{v}$ est le temps pour F de se déplacer de s_i à s_j . Donc, le temps moyen nécessaire pour p est $\frac{\sum_{1 \leq i, j \leq n} b_{ij} T_{ij}^p}{\sum_{1 \leq i, j \leq n} b_{ij}}$ où b_{ij} est la quantité moyenne de données collectées en se déplaçant de s_i vers s_j . Ensuite, le chemin p est amélioré par l'application de deux techniques d'optimisation :
- Remplacement d'arête (edge replacement): soit $s_i s_j$ et $s_l s_m$ deux arêtes dans p . soit p' le chemin modifié à partir de p en remplaçant $s_i s_j$ et $s_l s_m$ par $s_i s_l$ et $s_j s_m$. Si $T^{p'} < T^p$, alors p est remplacé par p' .
 - Ordonnancement de la séquence (sequence reordering): un nouveau chemin p' est construit par le déplacement de n'importe quel capteur s_i dans p de sa position originale vers une autre position. Si $T^{p'} < T^p$, alors p est remplacé par p' .
- 2) Optimisation de temps de transmission des données : Le champ de communication des capteurs peut over-lap, une politique d'allocation de temps (time allocation policy) ϕ_p est nécessaire pour assigner un temps de transmission aux capteurs. Le chemin p est divisé en m segments $\{\xi_1, \xi_2, \dots, \xi_n\}$ avec $\phi_p(s_i, \xi_j)$ représentant la portion du temps de transmission allouée au nœud s_i quand F se déplace le long du segment ξ_j . La figure 2.11 montre un exemple, où le sous-chemin (subpath) de u vers v est divisé en quatre (04) segments $\{\xi_1, \xi_2, \xi_3, \xi_4\}$.

--- chemin d'un data ferry

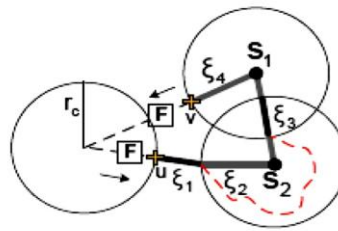


Figure 2.11. La décomposition de la trajectoire en segments [ZAM03].

Pour collecter toutes les données de chaque capteur s_i , F doit consacrer un temps suffisant pour communiquer avec s_i . S'il n'existe pas un temps suffisant, alors, le segment de s_i doit être allongé. Par exemple, dans la figure 2.11, le segment ξ_2 peut être allongé (la courbe en pointillés) afin d'augmenter le temps de communication avec s_i . Soit x_i un extra-temps de communication de F pour allonger ξ_j et t_i le temps de transmission original de s_i . Le problème est formulé par un problème de programmation linéaire qui minimise le total des extra-temps de transmission :

$$\left\{ \begin{array}{l} \min \sum_{j=1}^m x_j \\ \text{s.c.} \quad \frac{(t_i + \sum_{j=1}^m \phi_p(s_i, \xi_j) x_j) \cdot R}{\frac{|p|}{v} + \min \sum_{j=1}^m x_j} \geq b_i \end{array} \right.$$

Où R est le débit de données, et b_i est le besoin en bande passante de s_i .

b) La planification avec plusieurs data ferries :

Le travail effectué dans [ZAZ05] considère le cas de plusieurs data ferries. Etant donné n capteurs et m data ferries, le but est de trouver l'ensemble des chemins des data ferries pour visiter tous les capteurs tel que le délai moyen de message et le temps de transmission des données soient minimisés. Quatre types de solutions sont proposés :

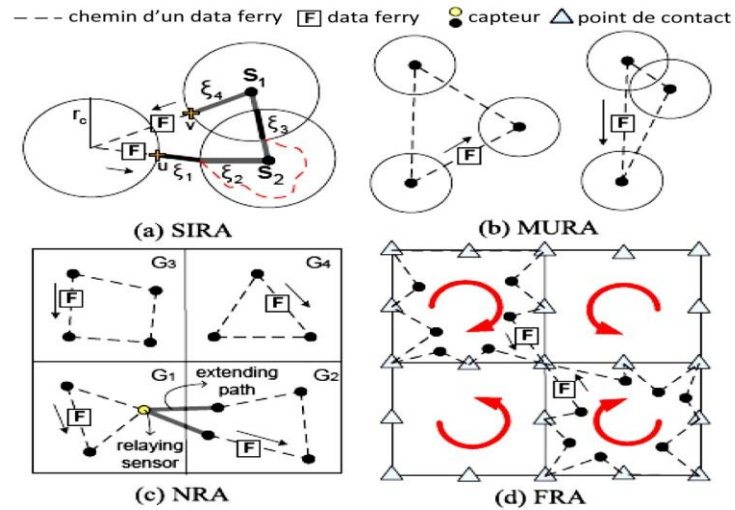


Figure 2.12. Planification du chemin (exemples pour les data ferries) [ZAZ05].

- **Algorithme de Single-route (SIRA):**

Tous les data ferries vont se déplacer le long d'un même chemin et sans aucune communication entre eux. La figure 2.12(a) montre un exemple de deux data ferries déplaçant le long d'un même chemin pour la collecte de données. Le principe de cet algorithme est le même que celui considéré dans [ZAM03], le problème est divisé en deux sous problèmes :

- 1) Optimisation de délai moyen : D'abord, pour n'importe quel chemin p calculé par un algorithme approximatif TSP, le délai de délivrer les données de s_i au s_j sur p est $T_{ij}^p = \frac{|p|}{2mv} + \frac{d_{ij}^p}{v}$. Le délai moyen de p est $T^p = \frac{\sum_{1 \leq i,j \leq n} w_{ij} T_{ij}^p}{\sum_{1 \leq i,j \leq n} w_{ij}}$, où w_{ij} est le poids assigné à chaque délai T_{ij}^p . Ensuite, les techniques d'optimisation locales : le remplacement d'arête (*edge replacement*) et l'ordonnancement de la séquence (*sequence reordering*) sont appliqués pour améliorer le délai moyen sur le chemin p .
- 2) Optimisation de temps de transmission des données : Le problème est formulé par un problème de programmation linéaire : où, y_i est la longueur de déplacement maximale (*the extra moving length*) des data ferries dans le champ de communication de s_i .

$$\begin{cases} \min \sum_{i=1}^n y_i \\ \text{s. c } R \cdot \frac{(2r_c + y_i)}{|p| + \sum_{j=1}^n y_j} \geq \frac{b_i}{m} \end{cases}$$

Dans la contrainte, la partie gauche représente le produit de « débit de données des data ferries » et « le temps de communication des data ferries alloué au s_i ». La deuxième partie signifie que le besoin en bande passante de s_i (représenté par b_i) est partagé entre le nombre de data ferries (m).

- **Algorithme de multi-route (MURA):**

Chaque data ferry va se déplacer le long d'un chemin différent et sans aucune communication entre eux. La figure 2.12 (b) montre un exemple de déplacement de deux ferries le long de deux chemins différents pour la collecte de données. Etant donné un ensemble de chemins P , une fonction de coût est définie par un 2-tuple $(E_1(P), E_2(P))$ pour évaluer la qualité de P , où $E_1(P)$ est le total estimé de surcharge des data ferries dans P , et $E_2(P)$ est le délai total estimé des messages encourus par P . Un chemin surchargé induit que les ferries à passer plus de temps à communiquer avec les nœuds capteurs ayant pour résultat des chemins plus longs et un délai important.

Initialement, à chaque capteur est assigné un ferry. Soit P l'ensemble des chemins courants, et n_i est le nombre de data ferries dans $p_i \in P$. Il existe quatre opérations pour réduire le nombre de data ferries et de raffiner l'ensemble des chemins P .

- 1) $Overlap(p_i, p_j)$: le chemin $p_j \in P$ est allongé par l'ajout d'un capteur de $p_i \in P, p_i \neq p_j$, tel que, le coût est minimisé.
- 2) $Merge(p_i, p_j)$: les chemins p_i et p_j sont combinés dans un nouveau chemin, et tous les ferries $n_i + n_j$ sont assignés au nouveau chemin.
- 3) $Merge^{minus}(p_i, p_j)$: la même chose que $Merge(p_i, p_j)$, avec diminution de nombre de ferries par un.
- 4) $Reduce(p_i)$: La diminution de n_i par un pour p_i , si $n_i > 1$.

Une opération parmi les quatre est choisie itérativement d'une manière gloutonne (*greedy manner*) pour minimiser le coût, jusqu'à la sélection de m chemins. Après l'obtention de m chemins, l'algorithme SIRA est appliqué pour optimiser chaque chemin.

- **Algorithme de node relaying (NRA)**

Dans cette méthode, chaque data ferry va se déplacer le long d'un chemin différent, et les capteurs statiques vont servir comme des nœuds de retransmission pour propager les données d'un chemin à un autre. Premièrement, la région d'intérêt est divisée en $c_1 \times c_2$ grilles, où $c_1 c_2 \leq m$, et chaque grille va être servie par un ferry qui voyage sur le chemin construit par SIRA. Supposant que les grilles G_s et G_d veulent échanger les données. Pour transmettre les données entre eux, il faut essayer de connecter G_s et G_d directement ou indirectement. Deux grilles peuvent être connectées par l'utilisation de l'opération $Overlap(p_i, p_j)$ définie dans MURA afin de trouver le nœud de retransmission (relaying node). La figure 2.12(c) montre un exemple, où il existe quatre grilles et quatre chemins. Ensuite, ces chemins vont être connectés par allongement d'un chemin vers un autre (extending path to another).

- **Algorithme de Ferry relaying (FRA)**

Similaire à NRA, la région d'intérêt est divisée en grilles, chaque grille peut être servie par un data ferry. Les data ferries se déplacent le long des chemins différents, mais ils peuvent communiquer entre eux pour l'échange des données. Les points de contact (contact points) sont désignés le long des bornes de la grille, comme montre la figure 2.12(d). Ces points de contact sont séparés par une distance de 0.5 de la longueur de la grille, donc chaque ferry a au plus huit (08) points de contact pour communiquer avec les autres ferries. Les data ferries de n'importe quelle deux grilles adjacentes vont se déplacer

dans la direction inverse de chacune des autres. Pour garantir que les data ferries puissent se rencontrer dans les points de contact, le chemin dans chaque grille est allongé par la connexion aux points de contact.

Les travaux présentés montrent qu'il y a trois schémas pour la communication des ferries :

- 1) Schéma sans interaction : les data ferries poursuivent le même chemin (SIRA), ou des chemins différents (MURA) mais sans aucune communication entre eux ;
- 2) Schéma d'interaction directe : les data ferries se déplacent le long des chemins différents, mais il existe une communication directe entre les data ferries pour l'échange des données (FRA);
- 3) Schéma d'interaction par le biais des nœuds relais : les data ferries se déplacent le long des chemins différents, et des capteurs statiques vont servir comme des nœuds relais pour la retransmission des données à travers les data ferries (NRA).

2.4.1.3. La planification Single-hop

La planification *single-hop* permet au ferry de contacter directement chaque capteur par une communication à un seul saut. Cette stratégie de planification optimise considérablement la consommation d'énergie et assure une fiabilité dans la collecte de données par élimination des communications multi-sauts entre les capteurs et le ferry. Des solutions sont proposées selon que le type de réseau soit homogène ou dispersé.

a) La planification single-hop dans un réseau homogène

Le travail de [MAY08] propose une planification du chemin de ferry pour voyager dans un RCSF tel que chaque capteur peut communiquer directement avec le ferry. Le chemin de data ferry est restreint à un ensemble de capteurs appelés les points d'élection (polling points). Le ferry n'est pas obligé de visiter chaque capteur dans le réseau pour la collecte de données. Il visite uniquement les points d'élections, les autres capteurs se situent à un seul saut des points d'élections (figure 2.13). Un ferry dans un polling point, il collecte les données de ce point et des nœuds voisins à un seul saut. Le but consiste dans la sélection des points d'élection tel que tous les capteurs sont voisins à un saut des points d'élection et le chemin de data ferry entre ces points soit minimisé.

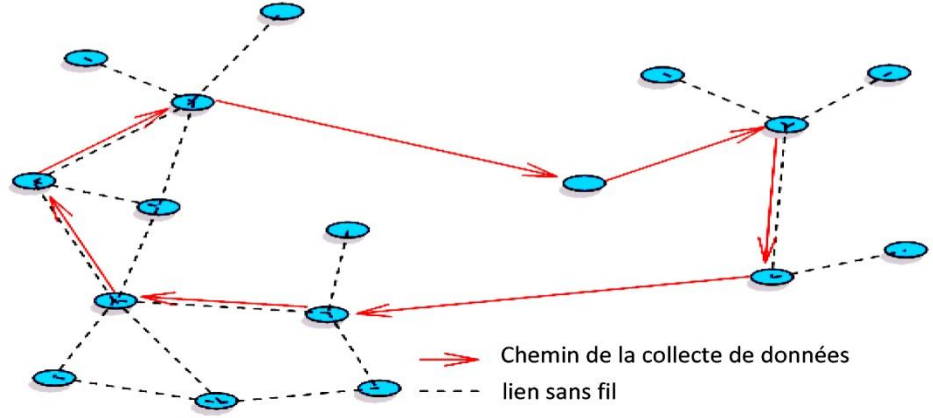


Figure 2.13. Le chemin de data ferry à travers les polling points [MAY08].

L'approche pour la planification de la trajectoire de ferry entre les points d'élection consiste à considérer un ensemble $\{l_1, l_2, \dots, l_k\}$ de points candidats d'élection (candidate polling points) qui contient toutes les positions de capteur. Un ensemble $N(l_i)$ de capteurs avec qui le ferry peut communiquer directement avec quand il arrive au point $l_i \in L$. Le but est de trouver un sous-ensemble de polling points $L' \subseteq L$ tel que chaque capteur appartienne au moins à un $N(l_i)$, $l_i \in L'$, et la longueur totale de chemin entre les polling points soit minimisée. Initialement, l'ensemble L' contient uniquement la station de base. Ensuite, une solution gloutonne (greedy solution) est proposée pour ajouter itérativement le polling point l_i dans L' avec un coût minimal de couverture (covering cost) $\tau(i)$, jusqu'à ce que tous les capteurs appartiennent au moins à un $N(l_i)$, $l_i \in L'$. Ici, $\tau(i)$ est défini par :

$$\tau(i) = \frac{\min \{d(l_i, l_j) | l_j \in L'\}}{|N(l_i) \cap \mu|},$$

où μ est l'ensemble de capteurs qui n'appartiennent à aucun $N(l_i)$, $l_i \in L'$, et $d(l_i, l_j)$ est la distance entre deux polling points l_i, l_j . Ensuite, un algorithme TSP est appliqué pour le data ferry afin de visiter tous les points dans L' . La référence [ZMY08] représente une extension du travail précédent en supposant que chaque capteur est équipé par une antenne, et le data ferry par deux antennes tel que le data ferry peut communiquer simultanément avec deux capteurs par la technologie SDMA (Space Division Multiple Access).

b) La planification single-hop dans un réseau dispersé

Soit un RCSF dispersé sur une zone géographique étendue, composé de sous-réseaux séparés l'un de l'autre. Un data ferry est responsable de la collecte de données par la visite d'un seul nœud dans chaque sous-réseau (figure 2.14). Le problème EM-TSP [FAN10]

consiste à optimiser la trajectoire de data ferry et la consommation d'énergie au niveau des capteurs. EM-TSP est une généralisation de problème TSP (problème NP-Complet). Basée sur les propriétés géométriques du réseau, une heuristique est proposée pour EM-TSP.

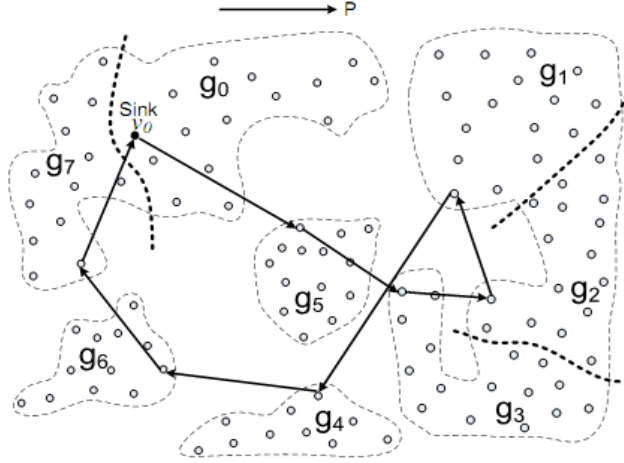


Figure 2.14. La planification de la trajectoire par EM-TSP.

Dans EM-TSP, l'heuristique appliquée sur l'ensemble des sous réseaux pour la planification de la trajectoire de mobile détermine d'abord pour le réseau l'ensemble des nœuds les plus proches du centre de gravité de chaque sous-réseau. Ensuite, un chemin TSP est calculé pour l'ensemble des nœuds candidats. Une amélioration est définie par l'examen des liens consécutifs de type $v_{pi} \rightarrow v_{pi+1} \rightarrow v_{pi+2}$, pour trouver un nœud v'_{pi+1} dans le même sous-réseau que v_{pi+1} avec $d(v_{pi}, v'_{pi+1}) + d(v'_{pi+1}, v_{pi+2}) < d(v_{pi}, v_{pi+1}) + d(v_{pi+1}, v_{pi+2})$, et remplacer v_{pi+1} par v'_{pi+1} . Ce processus est appliqué d'une manière itérative sur tous les liens consécutifs de chemin. Le chemin final est amélioré par la sélection des nœuds intérieurs comme des nœuds candidats (figure 2.15).

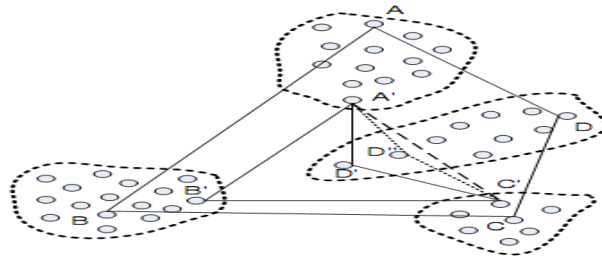


Figure 2.15. L'heuristique de EM-TSP pour la planification de la trajectoire.

Dans chaque sous réseau une contrainte sur l'énergie est définie, l'optimisation de l'énergie est assurée par la création des arborescences ayant comme racine les nœuds candidats pour chaque sous réseau. Un partitionnement est effectué sur les sous réseaux dans le cas où l'énergie de l'arborescence dépasse l'énergie limite.

2.4.2. La classe des travaux TSPN

Le problème TSPN est un variant de TSP, avec considération des champs de communication des capteurs qui sont modélisés par un disque (rayon de communication en cercle). La formulation classique de ce problème consiste en un ensemble de nœuds capteurs (modélisés par un disque) et un nœud collecteur qui doit assurer la collecte de données issues des capteurs. Le problème consiste à trouver le chemin optimal pour le nœud collecteur. Dans le cas classique de problème TSPN, les champs de communication sont de taille fixe et ils sont disjoints (il n'y a pas d'intersection entre eux). Dans la littérature d'autres considérations à ce problème ont été proposées pour : (1) le champ de communication des capteurs de taille variable (2) Les champs de communication des capteurs en intersection. Le premier problème est formalisé par un problème d'optimisation sur l'énergie de transmission des capteurs.

a) Le problème TSPN «TSP with neighborhood» :

Le problème TSPN est un problème NP-complet dont la résolution nécessite des heuristiques et des algorithmes approximatifs. Dans [NES06] et [NES07], les auteurs ont présenté un algorithme pour trouver le meilleur chemin de data ferry. Cet algorithme exige que les positions des capteurs soient connues. Il détermine d'abord l'ordre de visite des disques. Dans ce processus, quelques contraintes peuvent s'appliquer. Par exemple, les disques dont les capteurs correspondants sont sur le point d'épuiser leur énergie doivent être visités d'abord afin d'empêcher la perte de données. S'il n'y a aucune contrainte, alors la manière la plus intuitive pour ordonner les disques est de se baser sur l'ordre TSP de leurs points représentatifs. Le point représentatif d'un disque peut être choisi de différentes manières. Par exemple : un point aléatoire, un point central, ou le point le plus proche sur la circonférence au point de départ.

Une fois que l'ordre de visite est déterminé, l'algorithme calcule l'ensemble des points optimaux en conséquence. L'ensemble initial se compose de point de départ a_0 et le point représentatif a_i^0 du ième disque, C_i . Puis a_1^0 est mis à jour à a_1^1 par rapport à a_0, a_2^0 , et disque C_1 comme suit : si la ligne $a_0 a_2^0$ intersecte C_1 alors a_1^1 est tout point entre les intersections ; autrement, a_1^1 est un point sur la circonférence de C_1 tels que $|a_0 a_1^1| + |a_1^1 a_2^0|$ est réduit au minimum. Après que a_1^1 est calculé, a_2^0 est mis à jour à a_2^1 par rapport à a_1^1 et a_3^0 , et ainsi de suite. Finalement, a_n^0 est mis à jour par rapport à a_n^1 et a_0 et C_i . Une mise à jour itérative est appliquée sur l'ensemble des points jusqu'à trouver le

plus court chemin de ferry.

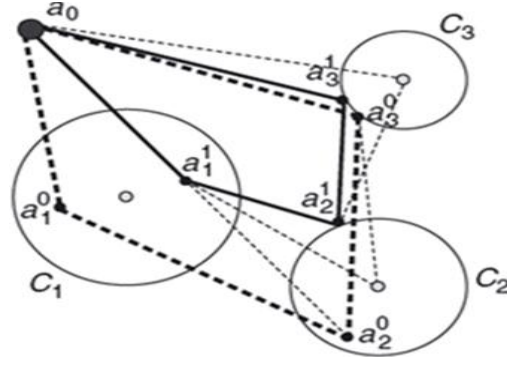


Figure 2.16. L'heuristique pour la résolution de problème TSPN.

2.4.2.1. TSPN avec problème d'optimisation

Ce travail vise principalement l'optimisation de l'énergie totale de transmission dans un RCSF par l'utilisation d'un data ferry, il formule l'énergie de transmission en fonction des données et des rayons de transmission au niveau de chaque capteur. Le problème d'optimisation consiste à trouver le chemin optimal pour le data ferry afin de minimiser l'énergie totale de transmission avec contrainte sur la latence.

Le problème est formulé de la manière suivante :

$$\begin{cases} \text{Min}_{p \in (\text{trajectoire})} E_T(p) = K \sum_{j=1}^N S_i(r_i(p))^a \\ \text{s. c} \quad D(P) \leq D_{max} \end{cases}$$

Avec $r_1, r_2, \dots, r_N$ se sont les rayons de transmission et $S_1, S_2, \dots, S_N$ les quantités des données à transmettre au niveau de chaque capteur. La solution à ce problème se base sur l'identification de la formule de rayon de transmission. La solution suggère que le data ferry doit être le plus proche des capteurs avec plus de données à collecter.

Une fois que les rayons de transmission sont définis pour chaque capteur en fonction des données à transmettre. Le RCSF résultant est constitué d'un ensemble de capteurs avec des rayons de transmission de taille différente. La planification de chemin optimal de data ferry pour la collecte de données est assurée par l'application des heuristiques définies pour la résolution de problème TSPN.

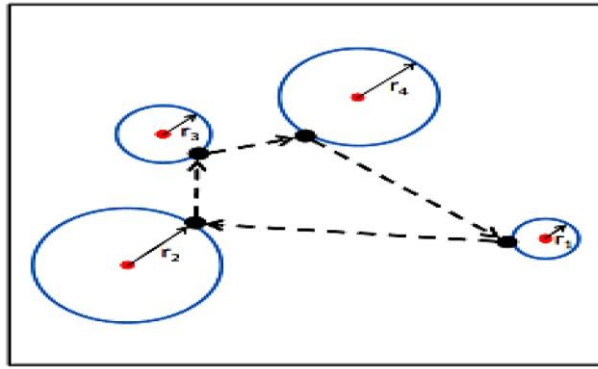


Figure 2.17. La planification TSPN avec des champs de communication de taille différente.

Dès lors, la partie centrale de ce problème d'optimisation est l'identification des rayons optimaux pour la transmission des données. Dans ce contexte, l'exigence que les rayons de transmission soient de même taille n'est pas une obligation. La taille d'un rayon de transmission est définie par la quantité des données dans les capteurs.

2.4.2.2. TSPN avec champ de communication en intersection

Selon la taille de réseau dense ou étendu, deux solutions sont proposées dans la littérature pour la planification de la trajectoire où les champs de communication en intersection. Deux stratégies différentes sont poursuivies dans chaque solution dans le but est l'optimisation de la trajectoire. Dans cette partie, nous allons exposer les deux solutions.

a) L'algorithme TSPNTOUR :

Dans le problème TSPN, il n'y a pas une intersection entre les champs de communication des capteurs. Ce travail considère un réseau avec des capteurs dont leurs champs de communication est en intersection. Un algorithme TspnTour a été proposé dans [DEE11] pour la planification de la trajectoire de data ferry par la sélection de l'ensemble maximal des capteurs avec leurs champs de communication disjoint l'un de l'autre. Ensuite, un chemin TSP qui passe par le centre de cet ensemble des capteurs est calculé. Afin d'assurer la couverture de tout le réseau ; le data ferry doit traverser le réseau en deux chemins : (i) un chemin extérieur est poursuivi dans le sens des aiguilles d'une montre pour assurer la couverture des capteurs qui sont en intersection avec la frontière supérieure. (ii) un chemin intérieur est poursuivi dans le sens inverse des aiguilles d'une montre pour assurer la couverture des capteurs qui sont en intersection avec la frontière inférieure. Le chemin final de data ferry est la combinaison de ces deux chemins (figure 2.18).

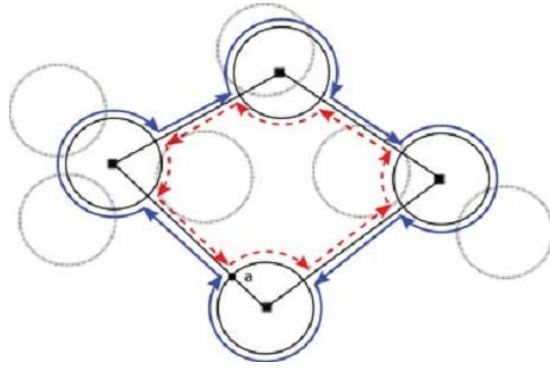


Figure 2.18. La planification de la trajectoire par TspnTour.

b) L'algorithme SparseTspnTour:

Dans la plupart des applications des data ferries de surveillance de l'environnement, le réseau est étendu et les capteurs sont loin l'un de l'autre. Dans de tels scénarios, l'algorithme TspnTour génère une redondance dans la planification par le parcours de la trajectoire deux fois. L'algorithme SparseTspnTour présente une amélioration qui consiste à un seul parcours de la trajectoire. L'algorithme détermine d'abord le chemin TSP qui passe par le centre de l'ensemble maximum des capteurs ayant leurs champs de communication disjoint l'un de l'autre. Ensuite, L'idée est de parcourir toute la frontière de champ de communication des capteurs appartenant à cet ensemble (figure 2.19).

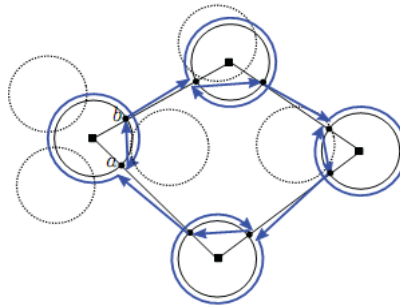


Figure 2.19. La planification de la trajectoire par SparseTspnTour.

Approches	Classe TSP			Classe TSPN	Densité réseau	Type réseau	Nombre de ferry	Paramètres à optimiser
	Ordonnancement	Adaptative	Single-hop					
PBS [YAO05]	X				Dense	homogène	un seul	Débordement tampon
MRME [YAO06]	X				Dense	homogène	un seul	Débordement tampon, messages urgents
Single-Ferry [ZAM03]		X			Dense	homogène	un seul	Latence, temps de transmission
SIRA [ZAZ05]		X			Dense	homogène	plusieurs	Latence, temps de transmission
MURA [ZAZ05]		X			Dense	homogène	plusieurs	Latence, temps de transmission
NRA [ZAM06]		X			Dense	homogène	plusieurs	Latence, temps de transmission
FRA [ZAZ03]		X			Dense	homogène	plusieurs	Latence, temps de transmission
Single-hop1 [MAY08]			X		Dense	homogène	un seul	Latence
Single-hop2 [ZMY08]			X		Dense	homogène	un seul	Latence
EM-TSP [FAN10]			X		Dense	dispersé	un seul	Latence, énergie des capteurs
OptTSPN [DEL09]				X	Dense	homogène	un seul	Latence, énergie des capteurs
TSPNTOUR [DEE11]				X	Dense	homogène	un seul	Latence
SparTSPNTOUR [DEE11]				X	Etendu	homogène	un seul	Latence

Tableau 2.1 .Récapitulatif des approches de la collecte de données basées TSP

2.4.3. Synthèse :

Les approches de la collecte de données par les data ferries basées TSP, considèrent des stratégies différentes pour la planification de la trajectoire. Tous les travaux visent à optimiser certains paramètres réseau (la latence, l'énergie, le temps de la collecte de données, etc). Sachant que le principal désavantage de l'approche de la collecte de données par les data ferries est l'augmentation de la latence, la majorité des travaux visent à optimiser la latence de la collecte de données.

Les approches pour de la collecte de données basées TSP sont regroupées en deux classes : les travaux qui modélisent le champ de communication par un point ponctuel (classe TSP), et ceux qui modélisent le champ de communication par forme géométrique (Classe TSPN).

Le problème de planification dans la classe des travaux TSP est formalisé par un problème d'optimisation et des heuristiques différentes sont proposées pour la résolution selon les caractéristiques de réseau et les paramètres à optimiser. Une classification est apportée à la classe des travaux TSP selon la stratégie poursuivie pour la planification : adaptative, ordonnancement et single-hop.

Dans la planification adaptative, des solutions dans cette famille considèrent le problème de la planification de la trajectoire de plusieurs data ferries. Ainsi, trois schémas de communication pour les data ferries sont définis : (i) schéma sans interaction, (ii) schéma d'interaction directe et (iii) schéma d'interaction par le biais des nœuds relais.

Pour la classe des travaux avec considération de champ de communication, elle considère les différentes variantes de problème TSPN. Chacune de ces variantes utilisent une stratégie différente. Notons que la majorité des travaux de la classe TSPN se basent sur les propriétés géométriques de réseau afin de suggérer des heuristiques pour la planification de la trajectoire de data ferry. La planification dans la classe des travaux TSPN est liée à la spécificité du champ de communication des capteurs. Deux cas de figures se présentent pour la planification : (i) la planification sans aucune communication entre les capteurs (les champs de communication sont disjoints les un des autres), (ii) la planification avec une communication entre les capteurs (les champs de communication des capteurs sont en intersection).

Le tableau 2.1 résume les différentes approches de la collecte de données basées TSP.

2.5. Conclusion

Dans ce chapitre, nous nous sommes intéressés à la planification de la trajectoire par les data ferries dans les RCSFs. Nous avons proposé une taxonomie pour les approches de la collecte de données dans les RCSFs selon la stratégie poursuivie pour la collecte de données : communication directe ou basée points de rendez-vous. Ainsi, nous avons décrit les approches de la planification de la trajectoire basées TSP pour les deux classes TSP et TSPN avec une étude et synthèse des travaux de chaque classe.

Les approches de la planification de la trajectoire basées TSP sont destinées aux applications de surveillance constituées d'un ensemble de capteurs statiques positionnés sur des positions bien déterminées et une entité mobile (drone, véhicule) se déplace pour la collecte de données et qui modélisent le champ de communication par une forme géométrique. Nous proposons dans le prochain chapitre une nouvelle solution pour la planification de la trajectoire de data ferry pour un environnement réel où nous proposons un modèle de communication probabiliste.

Chapitre 3 PSFPP : La solution pour la planification de la trajectoire de data ferry proposée

3.1. Introduction

Dans le chapitre précédent, nous avons présenté un état de l'art sur les approches de la planification de la trajectoire de data ferry et principalement les approches basées TSP. Dans la littérature, les approches de la planification de la trajectoire, avec considération du champ de communication, modélisent le champ de communication par une forme circulaire appelée communément le modèle de communication binaire ou le modèle disque. Ce modèle permet de ramener le problème de la planification de la trajectoire de data ferry à un problème géométrique. Bien que le modèle disque simplifié l'analyse, il est considéré comme étant trop simpliste et ne reflète pas fidèlement la réalité. En effet, une description plus réaliste du réseau réel de capteurs est que le champ de communication des capteurs diminuant avec le temps et la distance qui les sépare.

Dans ce chapitre, nous considérons le scénario suivant : soit une zone d'intérêt (un champ de bataille par exemple) avec des capteurs statiques déployés sur des positions prédéterminées pour surveiller cette zone. La collecte des données observées par les capteurs se fait via un véhicule automatique non-piloté (terrestre ou aérien) en mouvement autour de la zone surveillée. Le problème considéré ici est : comment planifier la trajectoire du data ferry (véhicule non-piloté) tout en considérant un modèle de communication probabiliste. Pour résoudre ce problème nous présentons notre solution PSFPP (Probabilistic Solution for Ferry Path Plan). Nous commençons d'abord par développer un modèle de prédiction des coûts de communication. Ensuite, nous reformulons le problème de la planification de la trajectoire du data ferry tout en considérant notre modèle de prédiction des coûts de communication. Finalement, nous présentons notre proposition pour la résolution de ce nouveau problème.

3.2. Formulation du problème de la planification de la trajectoire de data ferry

Avant de présenter la nouvelle formulation du problème de la planification de la trajectoire de data ferry, nous rappelons dans cette section, sa formulation d'origine basée sur le modèle de communication binaire. La formulation classique de la planification de la trajectoire de data ferry est comme suit : étant donné un ensemble de N capteurs déployés dans une zone d'intérêt ; quel chemin doit emprunter un élément mobile qui se déplace dans la zone d'intérêt pour collecter les données de chaque capteur ?. L'objectif est la planification de la trajectoire de data ferry sous contrainte de la minimisation de l'énergie des capteurs avec minimisation de la latence de collecte de données.

Ce problème de planification est une variante du problème de voyageur de commerce (TSP) [GIL92, GER94, NIC76]. Le problème consiste à optimiser la trajectoire du data ferry afin qu'il passe sur le champ de communication de chaque capteur pour la collecte des données. Dans un RCSF, le champ de communication d'un capteur est modélisé par un disque. Le problème avec considération de champ de communication des capteurs est une classe des problèmes TSPN (*Travelling Salesperson Problem with Neighbourhood*). La figure 3.1 illustre ce problème.

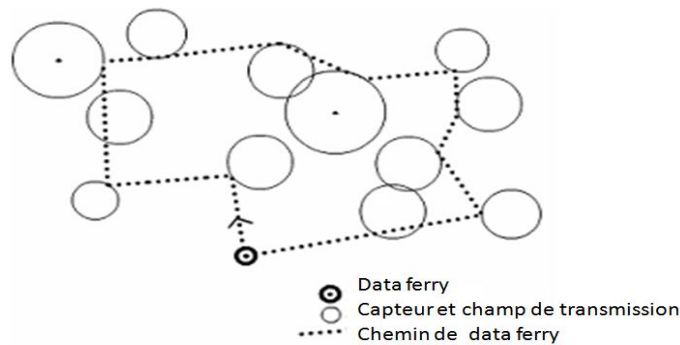


Figure 3.1. Planification de la trajectoire de data ferry.

3.2.1. La formulation mathématique de problème TSPN :

Les paramètres à optimiser pour ce problème classique de planification sont :

- La consommation d'énergie totale du réseau : énergie de transmission au niveau des capteurs et énergie de réception au niveau de data ferry ;
- La longueur totale de chemin de data ferry par conséquent la latence de la collecte de données.

3.2.1.1. Optimisation énergétique :

Nous adoptons le modèle général de la consommation d'énergie de Rodoplu et Meng présenté dans [ROD99]. Supposons que les N capteurs déployés dans la zone d'intérêt ont la même quantité de données à transmettre S , l'énergie de transmission nécessaire pour transmettre S bits à travers une distance r_i est donnée par : $E_t(r_i) = K S r_i^\alpha$ (K une constante, $2 \leq \alpha \leq 6$), et l'énergie de réception $E_r = S b$ (b une constante représente l'énergie consommée par les circuits électroniques de data ferry due à la réception d'un seul bit). La forme générale de l'énergie totale de la collecte de données (la transmission de S bits à partir de chaque capteur vers le ferry) est : $E_c(r_i) = E_t(r_i) + E_r$. Donc le problème d'optimisation de l'énergie est formulé par :

$$\text{Min } \sum_{i=0}^N E_c(r_i) = \text{Min } S \sum_{i=0}^N (K r_i^\alpha + b)$$

3.2.1.2. Optimisation de la trajectoire :

Soit la zone d'intérêt P , et $n+1$ sous ensembles $\{F, S_1, S_2, \dots, S_n\} \in P$. F représente la position de départ de data ferry et $\{S_i\}_{i=1}^n$ représentent les points de champ de transmission des nœuds capteurs. $F = \{a_0\}$, $S_1 = \{a_{11}, a_{12}, \dots, a_{1|S_1|}\}$, $S_2 = \{a_{21}, a_{22}, \dots, a_{2|S_2|}\}$, $S_n = \{a_{n1}, a_{n2}, \dots, a_{n|S_n|}\}$ et en général, $S_i = \{a_{ij}\}_{j=1}^{|S_i|}$. Le problème consiste à construire un chemin P' inclus dans P qui contient au moins un point de chaque sous ensemble. L'objectif est de minimiser la longueur de chemin P' . Le problème est formulé de la manière suivante :

Min $d_{\pi\varphi}$ Avec :

- $d_{\pi\varphi} = \sum_{i=0}^n d_{a_{\pi(i)\varphi(\pi(i))} a_{\pi((i+1)\bmod(n+1))\varphi(\pi((i+1)\bmod(n+1))})}$
- π est une permutation sur n
- $\pi : \{0, \dots, n\} \rightarrow \{0, \dots, n\}$
- $\pi(i)$ indique le point à la $i^{\text{ème}}$ position dans le chemin
- φ est une permutation sur $\pi(i)$
- $\varphi : \{1, 2, \dots, |S_{\pi(i)}|\} \rightarrow \{1, 2, \dots, |S_{\pi(i)}|\}$

Notre objectif est de reformuler le problème de la planification de la trajectoire de data ferry tout en considérant un modèle de communication réaliste. Dans la section suivante, nous discutons le modèle de communication binaire et ses limites et nous développons un nouveau modèle de communication probabiliste.

3.3. Modèle de connectivité probabiliste et les coûts de communication

Il est d'usage de supposer un modèle de communication (connectivité) binaire selon lequel deux capteurs sont capables de communiquer directement si chacun d'eux est à l'intérieur du rayon de communication (R_c) de l'autre (voir Figure 3.2). Bien que ce modèle soit simple et qu'il considère le problème de connectivité des réseaux, principalement, d'un point de vue géométrique ; ce modèle n'est pas réaliste. En effet, les études empiriques montrent qu'il n'existe pas de frontière nette entre les communications réussies et échouées tel que suggéré par le modèle de communication binaire.

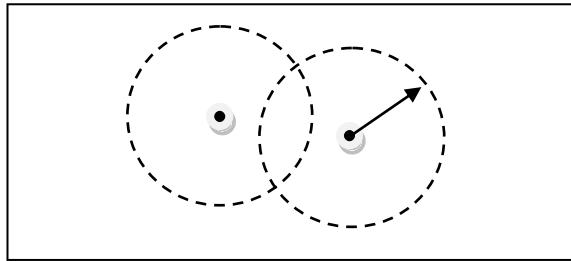


Figure 3.2. Le modèle de communication binaire.

Dans la pratique, l'affaiblissement subi par un signal sans fil à une distance donnée est décrit par le *pathloss* alors que le *shadowing* décrit les petites variations aléatoires dans la force du signal. Les mesures empiriques ont montré que le shadowing est une variable aléatoire distribuée qui suit une loi normale avec un écart type σ_ϵ et une moyenne nulle. En raison des caractéristiques uniques de chaque environnement, la plupart des modèles radio de propagation utilisent une combinaison de méthodes analytiques et empiriques. Un des modèles radio de propagation le plus courant est le modèle *log-normale shadowing pathloss* [RAP01] qui est donnée par :

$$PL(d) = PL(d_0) + 10\gamma \log_{10}(d / d_0) + \epsilon \quad (\text{III.1})$$

Où d est la distance qui sépare le transmetteur du récepteur, d_0 est une distance de référence, γ est l'exposant du pathloss (vitesse à laquelle le signal s'affaiblit), et ϵ est une variable aléatoire Gaussienne de moyenne nulle avec une déviation standard σ_ϵ (en dB) qui représente l'effet du shadowing (les variations dues aux effets de l'environnement).

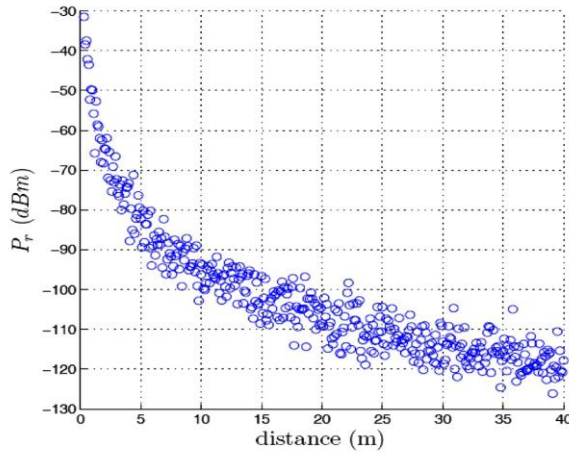


Figure 3.3. Modèle de propagation avec $\gamma = 2$, $\sigma_\epsilon = 4$, $P_t = 0$ dBm.

La force du signal à la réception Pr à une distance d c'est la puissance de sortie de l'émetteur moins $PL(d)$. Formellement :

$$Pr(d) = P_t - PL(d) = P_t - PL(d_0) - 10\gamma \log_{10}(d/d_0) - \epsilon \quad (\text{III.2})$$

La figure 3.3 montre le modèle de propagation analytique avec $\gamma = 2$, $\sigma_\epsilon = 4$, $PL(d_0)=55\text{dB}$, $d_0=1$ et une puissance de sortie $P_t = 0$ dBm (par exemple le Chipcon CC2420 IEEE 802.15.4, 2.4 GHz).

A partir de l'équation précédente, nous avons $Pr(d)$ qui suit la loi normale $N(P_t - PL(d_0) - 10\gamma \log_{10}(d/d_0), \sigma_\epsilon)$ [RAP01]. Étant donné que $Pr(d)$ est Gaussienne, la probabilité d'une communication réussie entre deux capteurs s_i et s_j situés à une distance d l'un de l'autre est :

$$Proba[Pr(d) > Seuil_{min}] = Q((Seuil_{min} - (P_t - PL(d_0) - 10\gamma \log_{10}(d/d_0))) / \sigma_\epsilon) \quad (\text{III.3})$$

Où $Seuil_{min}$ représente la puissance du signal acceptable minimum et Q est la fonction de distribution cumulative complémentaire (CCDF) d'une gaussienne standard :

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{+\infty} e^{-\frac{t^2}{2}} dt \quad (\text{III.4})$$

Par conséquent, le nombre prévu de retransmissions pour une communication réussie entre s_i et s_j est :

$$NR(d) = 1 / Proba[Pr(d) > Seuil_{min}] \quad (\text{III.5})$$

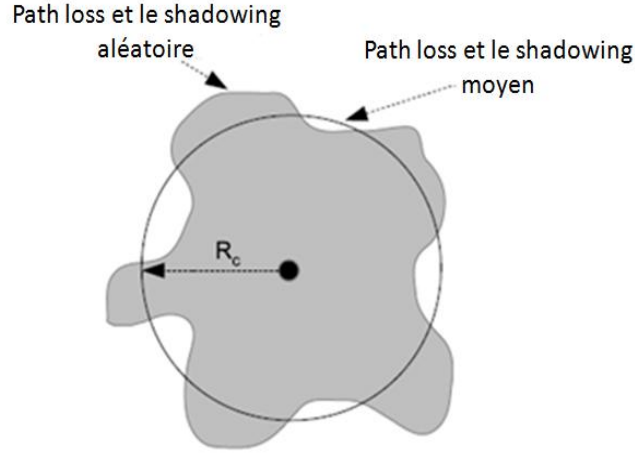


Figure 3.4. Le modèle de connectivité.

La figure 3.4 montre le modèle de connectivité lié à cette formulation. Nous voyons clairement que certaines zones à l'intérieur du disque de communication ont reçu une puissance inférieure à $Seuil_{min}$ alors que d'autres zones à l'extérieur du disque de communication ont reçu une puissance supérieure à $Seuil_{min}$. Comme mentionné auparavant, ce modèle montre bien le fait qu'il n'y ait pas une frontière nette entre les communications réussies et échouées.

Sur la base du modèle de connectivité décrit précédemment, nous définissons le coût de communication d'un lien ij (notée CL_{ij}) comme étant le nombre attendu de retransmissions nécessaires pour une communication réussie entre les nœuds s_i et s_j .

$$CL_{ij} = NR(\|ij\|) = \frac{1}{Proba[P_r(\|ij\|) > Seuil_{min}]}$$

Où $\|ij\|$ est la distance Euclidienne entre deux points i et j . Dans la section suivante, nous utilisons ce modèle pour redéfinir le problème de planification de la trajectoire de data ferry.

3.4. Nouvelle formulation du problème de planification de la trajectoire de data ferry

Après avoir détaillé le modèle de connectivité, nous reformulons le problème de la planification de la trajectoire de data ferry. Nous voulons optimiser simultanément la durée de vie du réseau et la latence. Étant donné que la transmission consomme plus d'énergie que toute autre tâche, nous devons nous assurer que les capteurs déployés aient des liens de communication fiables et que le nombre total de retransmissions inutiles soit

minimisé. La latence de la collecte de données par l'élément mobile est influencée par la distance parcourue par le data ferry, ainsi la minimisation de la distance de déplacement de data ferry diminue la latence de la collecte de données.

3.4.1. Modèle réseau et hypothèses

Nous considérons une zone d'intérêt représentée par un modèle réseau simple et discret : une grille bidimensionnelle de dimension $X \times Y$ cellules. N capteurs sont déployés d'une manière aléatoire et un seul data ferry est employé pour la collecte de données sur le RCSF. La figure 3.5 montre une représentation schématique de notre modélisation du réseau de capteurs.

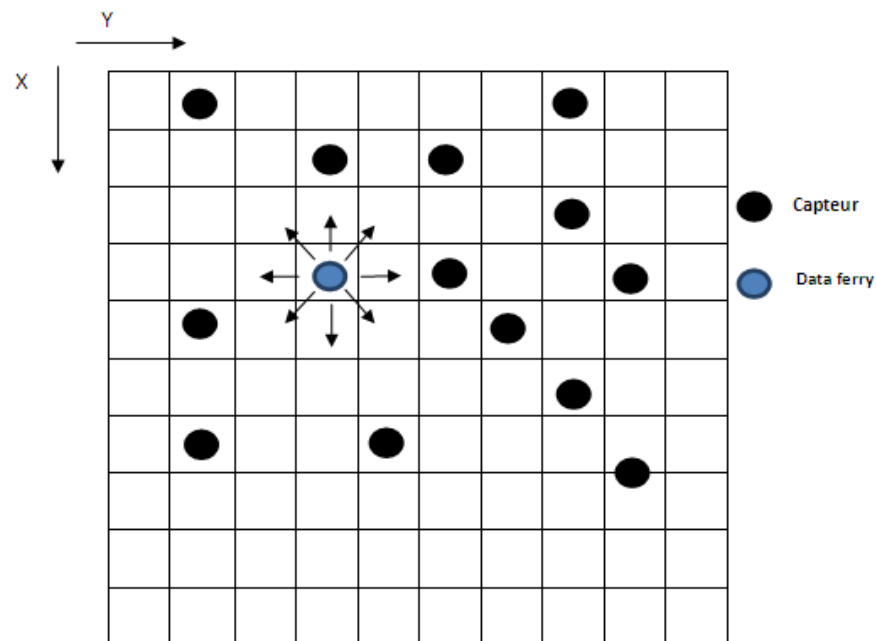


Figure 3.5. Une modélisation en grille bidimensionnelle.

Par rapport à l'environnement considéré, nous posons les hypothèses suivantes :

- Le data ferry et les nœuds capteurs utilisent le même modèle de communication, à savoir le modèle de communication probabiliste ;
- Les N capteurs déployés dans la scène sont statiques ;
- Chaque capteur est positionné au centre d'une cellule de taille unitaire, chaque cellule comporte un et un seul capteur ;
- Le centre de chaque cellule est un emplacement possible pour un capteur, l'ensemble des positions des capteurs déployés est noté D ;
- Le data ferry se déplace dans la grille à travers un ensemble de cellules noté T dans toutes les directions et à une vitesse constante v pour la collecte de données.

3.4.2. Formulation du problème

Conceptuellement, le problème d'optimisation adressé dans cette section est comme suit : soit $T_0 \in T$ la position de départ du data ferry, le problème est de construire une route $T' \subseteq T$ et qui démarre de T_0 et qui revient à T_0 telle que cette route minimise le coût de communication total (nombre de retransmissions) entre le data ferry et les N capteurs déployés. L'objectif est de minimiser la longueur de cette route avec optimisation des coûts de communication.

Formellement, le problème de la planification de la trajectoire d'un data ferry avec prise en compte de notre modèle de communication, peut être défini comme suit :

$$\min \sum_{i \in T, j \in D} CL_{ij}$$

$$\min \sum_{i, j \in T} \|ij\|$$

Le problème consiste donc à :

1. Minimiser le coût des communications (nombre de retransmissions) entre le data ferry et les capteurs, ceci permet d'avoir une meilleure qualité de communication avec optimisation de la consommation énergétique ;
2. Minimiser la distance euclidienne entre les différentes positions du data ferry, ce qui minimise la longueur de la route T' et par conséquent minimise aussi la latence de collecte de données.

Ce problème est un problème d'optimisation multi-objectif. Le but que l'on se fixe dans la résolution de ce problème d'optimisation est de minimiser "au mieux" les différents objectifs.

3.5. Notre approche de résolution

Le nouveau problème formulé est loin d'être une simple instance du problème classique TSP. Pour résoudre ce nouveau problème, nous procédons par étape. Dans la suite de cette section, nous détaillons chaque étape, avec les concepts et les méthodes utilisées pour aboutir au résultat escompté. Nous commençons d'abord par définir la fonction score introduite. Ensuite, nous expliquons les étapes de résolution.

3.5.1. Définition de la fonction score

Dans la solution recherchée, il faut absolument éviter de construire une route avec des liens de communication de mauvaise ou très mauvaise qualité. En d'autres termes, il faut éviter que le coût de communication d'un lien devienne très grand. Pour cela, nous introduisons une nouvelle fonction appelée score. Cette fonction est définie comme suit :

$$score(d) = \begin{cases} \frac{1}{CL(d)} & \text{si } d \leq d_s \\ 0 & \text{sinon} \end{cases}$$

Par définition, le score d'un lien est l'inverse du coût de communication d'un lien (CL) défini précédemment en ignorant certains liens. Nous introduisons une distance seuil d_s pour éviter les liens de communication de mauvaise qualité. Donc, si un capteur est très loin par rapport au data ferry, le score du lien de communication est 0, ceci permet d'éviter de communiquer avec le capteur en question depuis une position courante avec une mauvaise liaison du data ferry. En d'autres termes, pour communiquer avec ce capteur, le data ferry est obligé de se rapprocher du capteur pour avoir un lien de communication de meilleure qualité (avec un bon score).

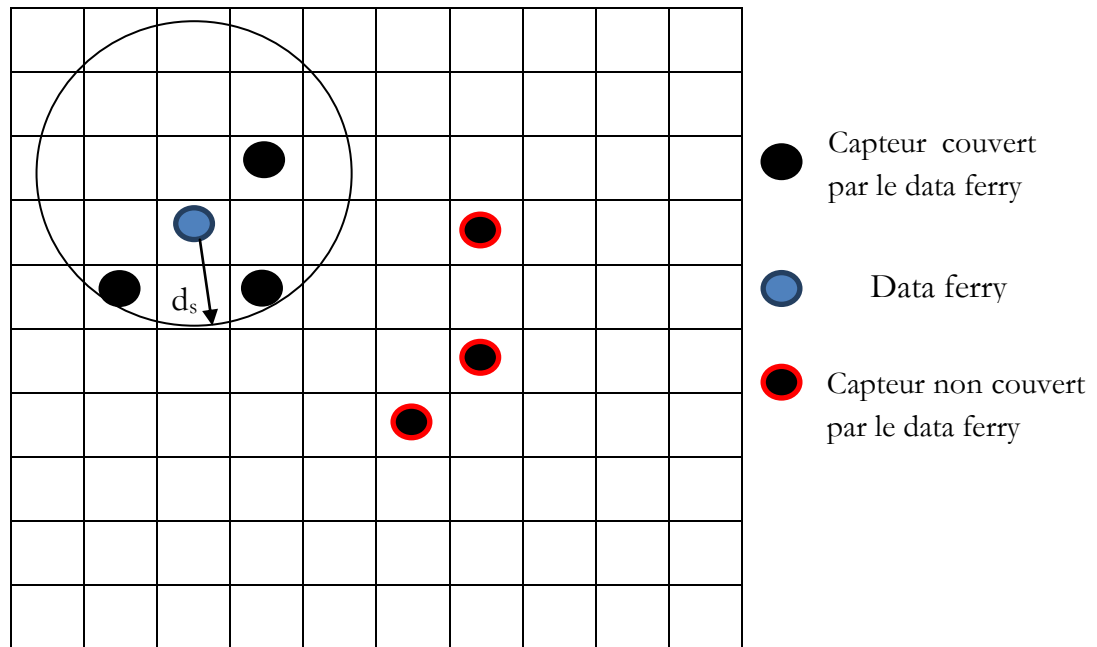


Figure 3.6. illustration schématique de la fonction score.

La figure 3.6 illustre le principe précédent. Les liens de communication des capteurs situés à une distance inférieure à d_s par rapport au data ferry sont considérés dans les solutions candidates. Par contre, les liens de communication des capteurs éloignés (situés à une distance supérieure à d_s) ne sont pas considérés (score nul). Par abus de langage,

nous appelons d_s le rayon de couverture du data ferry. Les capteurs situés à une distance inférieure à d_s sont dits couverts par le data ferry. Par contre, les capteurs situés au delà de cette distance ne sont pas couverts par le data ferry (figure 3.6). Pratiquement, la distance seuil d_s sera paramétrée en fonction des caractéristiques de l'entité mobile (data ferry).

Dans la suite de cette section, nous détaillons les étapes de résolution.

3.5.1.1. Étape 1 : Construction des vecteurs de couverture

Pour chaque position $T_i \in T$, nous définissons un vecteur appelé vecteur de couverture du collecteur mobile (VCC) qui indique les capteurs qui sont couverts (au sens de la fonction score) par le data ferry et la somme de leurs scores. Une définition formelle de ce vecteur est donnée comme suit :

$$VCC[i] = \begin{cases} (x, y) & \text{Pour } i = 0 \wedge 1 ; \text{ la position du data ferry dans la grille} \\ \sum_{j=3}^{N+3} b_j \cdot score(d_j) & \text{Pour } i = 2; \text{ la somme des scores des capteurs couverts} \\ \sum_{j=3}^{N+3} b_j \cdot CL(d_j) & \text{la somme des coûts des capteurs couverts par le data ferry} \\ b_i = 0 \vee 1 & \text{Pour } i = 4 \text{ à } N + 4; b_i = 1 \text{ capteur couvert sinon non couvert} \end{cases}$$

La forme générale du vecteur VCC est la suivante :

x	y	$\sum score$	$\sum coût$	b_4	b_5			b_{N+4}
---	---	--------------	-------------	-------	-------	------	-------	-----------

Le vecteur VCC est de dimension $N+4$, les deux premières cases du vecteur : représentent la position (x,y) du collecteur dans la grille, la troisième et la quatrième case de vecteur représente $\sum score$ et $\sum coût$ des capteurs qui sont dans le rayon de couverture du collecteur qui se trouve dans la position (x,y). Les N dernières cases du vecteur VCC : incluent une valeur binaire (0 ou 1) qui indique si le capteur est dans le rayon de couverture du collecteur ($b_i=1$), sinon ($b_i=0$). Avec N le nombre des capteurs déployés dans la zone d'intérêt.

3.5.1.2. Étape 2 : Construction de la matrice de couverture de la grille « MCG »

Après avoir défini pour chaque position $T_i \in T$ un vecteur « VCC », la matrice de couverture de la grille « MCG » est construite à base de ce vecteur. Elle englobe pour les différentes positions T du collecteur mobile dans la grille leurs vecteurs de couverture « VCC ». Une définition formelle de cette matrice est donnée comme suit :

$$\text{MCG} = \{VCC_{(x,y)} : \text{vecteur de couverture du collecteur à la position } (x,y) / (x,y) \in T\}$$

Dans le cas où T peut prendre l'ensemble des positions possibles dans la grille, la forme générale de la matrice MCG est la suivante :

0	0	$\sum \text{score}_{(0,0)}$	$\sum \text{coût}_{(0,0)}$	1		0	1
0	1	$\sum \text{score}_{(0,1)}$	$\sum \text{coût}_{(0,1)}$	1		1	1
.	.			1		0	0
0	Y	$\sum \text{score}_{(0,Y)}$	$\sum \text{coût}_{(0,Y)}$	0		0	1
.	.						
X	0	$\sum \text{score}_{(X,0)}$	$\sum \text{coût}_{(X,0)}$	1		0	0
.	.						
X	Y	$\sum \text{score}_{(X,Y)}$	$\sum \text{coût}_{(X,Y)}$	1		1	0

La matrice de couverture de la grille « MCG » est de dimension $(X*Y, N+4)$, avec X nombre de ligne de la grille, Y nombre de colonne de la grille et N nombre de capteurs déployés dans la zone d'intérêt. Cette matrice englobe les différentes positions de l'entité mobile dans la grille de la position $(0,0)$ à la position (X,Y) .

3.5.1.3. Étape 3 : Détermination des sous-ensembles de couverture

L'objectif de cette étape est de déterminer les sous-ensembles $T' \subseteq T$ qui couvrent l'ensemble des capteurs déployés. Nous appelons ces sous-ensembles les sous-ensembles de couverture. En d'autres termes, nous essayons de trouver toutes les routes $T' \subseteq T$ telles que chaque route assure la couverture de tous les capteurs. Si le collecteur mobile empreinte n'importe quelle route T' , nous sommes certains qu'il peut communiquer avec tous les capteurs déployés. Bien entendu, chaque position $p \in T'$ permet au collecteur de communiquer avec quelques capteurs.

A partir de la matrice MCG, nous générons les différents sous-ensembles de couverture. Pour cela, nous procédons de la manière suivante :

1. **Phase 1** : Elimination des positions qui n'assurent pas la couverture d'aucun capteur : dans cette phase, nous supprimons simplement les positions $T_i \in T$ qui ne couvrent aucun capteur.
2. **Phase 2** : Classification des vecteurs qui couvrent les mêmes capteurs : dans cette phase, nous regroupons les vecteurs VCC qui couvrent les mêmes capteurs. Pour cela, nous définissons des classes C_i qui contiennent les vecteurs VCC qui couvrent les mêmes capteurs :

$$C_i = \{VCC_k^i \text{ qui couvrent les mêmes capteurs } \setminus i : \text{indice de la classe et } k : \text{la taille de la classe}\}$$

Nous associons à chaque classe C_i un vecteur représentatif de la classe nommé V_i . Il contient l'indice de la classe et N valeurs binaires qui représentent l'ensemble des capteurs qui sont couverts par cette classe. Le vecteur V_i est de la forme suivante :

Indice i de la classe C_i	Les capteurs couverts par cette classe i
-------------------------------	--

3. **Phase 3** : Réduction de la matrice MCG à la matrice réduite MCGR. Nous réduisons la matrice MCG à la matrice MCGR qui contient l'ensemble des vecteurs représentatifs des classes :

$$MCGR = \{V_i : \text{vecteur représentatif de chaque classe } C_i / i \in \{0, \dots, \text{nombre de classe}\}\}$$

4. **Phase 4** : Nous générons à partir de la matrice MCGR les combinaisons de vecteurs de la forme $V_1 \oplus V_2 \oplus \dots \oplus V_k$ qui assurent la couverture de réseau :

Pour la détermination des ces combinaisons, nous appliquerons un algorithme approximatif de SSP (Sub-Set Sum Problem) dont la formulation du problème : consiste à trouver à partir d'un ensemble d'entiers $\{a_0, a_1, \dots, a_n\}$ et une valeur entière K , le maximum de combinaisons dont leurs somme vaut K . Ce problème est un problème NP-Complet. La stratégie poursuivie pour la résolution du problème de la couverture réseau par application d'un algorithme SSP se résume en :

-La subdivision de la partie binaire de chaque vecteur représentatif en digits (groupement de 4bits), le vecteur V_i devient de la forme suivante :

V_{i1}	V_{i2}				$V_{iN/4}$
----------	----------	--	--	--	------------

Avec $V_{i1}, V_{i2}, \dots, V_{iN/4}$ des vecteurs de 4 bits, et N nombre des capteurs dans le réseau. Nous choisissons un groupement de 4bits au lieu des groupements de 16, 32, ..., 2^k bits afin d'augmenter le nombre des solutions.

-La construction d'une matrice à base de cette subdivision.

-L'application de l'algorithme SSP sur chaque colonne j de la matrice construite, l'idée est de déduire le maximum des combinaisons sur chaque colonne j dont la somme vaut $(1111)_2 = F_{16}$. Formellement : $\max \sum_{i=1}^m V_{ij} = (1111)_2 = F_{16}$, avec $1 \leq j \leq N/4$ et m :nombre des vecteurs représentatifs. Intuitivement, nous allons faire ressortir pour chaque colonne le maximum de combinaisons qui assure la couverture des capteurs par groupement de quatre : $\{s_1, s_2, s_3, s_4\}$ pour la première colonne, $\{s_5, s_6, s_7, s_8\}$ pour la deuxième colonne, ..., $\{s_{j-3}, s_{j-2}, s_{j-1}, s_j\}$ pour la j^{ème} colonne.

-La combinaison entre les sous-ensembles de chaque colonne, pour en déduire les sous-ensembles de la couverture de réseau.

5. **Phase 5** : Génération des listes des positions. Ensuite, nous appliquerons les combinaisons générées dans la phase précédente sur l'ensemble des classes pour déduire les listes des différentes positions du nœud mobile pour la couverture de réseau.

Le vecteur de couverture de réseau est de la forme :

Les listes des positions x_i, y_i	$\sum$ coût des x_i, y_i	111111.....111
-------------------------------------	----------------------------	----------------

3.5.1.4. Etape 4 : Détermination de la longueur de chemin de chaque sous-ensemble

Nous appliquons un algorithme de TSP sur les listes des positions des différents sous-ensembles pour en déduire les distances minimales de chaque sous ensemble :

TSP(sous-ensemble) → distance minimale ;

A l'issue de cette phase, nous élaborons ainsi un tableau contenant la liste des positions (x_i, y_i) , $\sum$ coût et $\sum$ distance de la forme suivante :

Liste des positions (x_i, y_i)	$\sum$ coût	$\sum$ distance

Tableau 3.1. La forme du tableau des solutions.

La détermination d'un sous-ensemble de l'ensemble des solutions est réalisée dans l'étape suivante.

3.5.1.5. Etape 5 : Définition de la fonction objectif

La résolution d'un problème d'optimisation multi-objectif, produit une grande quantité de solutions. Ces solutions, comme on peut s'en douter, ne sont pas optimales, au sens où elles ne minimiseront pas tous les objectifs du problème.

Un concept intéressant, qui nous permettra de classer les solutions obtenues, est le compromis. En effet, les solutions que l'on obtient lorsque l'on a résolu le problème sont des solutions de compromis. Elles minimisent un certain nombre d'objectifs tout en dégradant les performances sur d'autres objectifs.

Lorsque nous résolvons notre problème d'optimisation, nous obtenons une multitude de solutions. Seul un nombre restreint de ces solutions va nous intéresser. Pour qu'une solution soit intéressante, il faut qu'il existe une relation de dominance entre la solution considérée et les autres solutions, dans le sens suivant [YAN02]:

On dit que la solution S_1 domine la solution S_2 si :

- S_1 est au moins aussi bon que S_2 dans tous les objectifs, et,
- S_1 est strictement meilleur que S_2 dans un au moins un objectif.

Les solutions qui dominent les autres mais ne se dominent pas entre elles sont appelées **solutions optimales au sens de Pareto** (ou **solutions non dominées**). Ces solutions forment ce que l'on appelle la **surface de compromis** (ou **front de Pareto**). La figure 3.7 montre la forme de l'ensemble des solutions optimales au sens de Pareto pour les différents problèmes d'optimisation à deux objectifs. La figure 3.7 (a) correspond à notre cas d'étude où les deux objectifs à minimiser est : $\sum$ coût et $\sum$ distance.

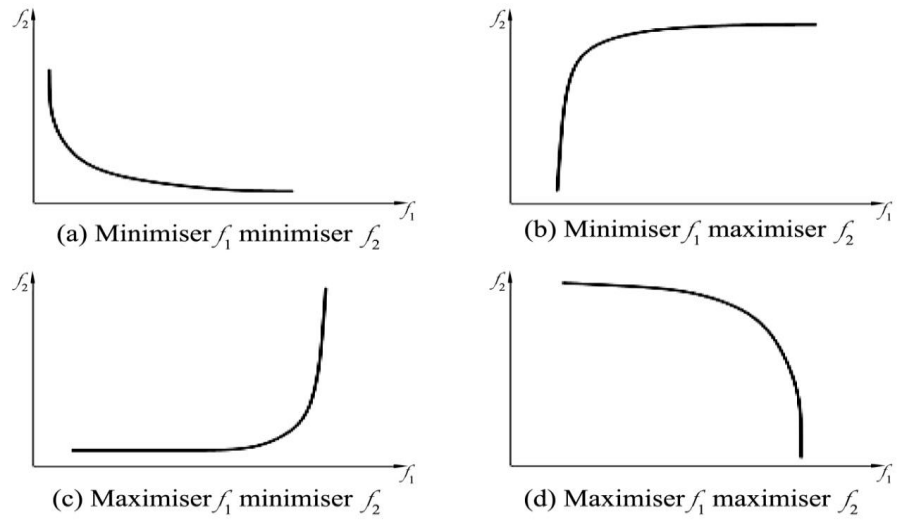


Figure 3.7. Formes de l'ensemble des solutions optimales au sens de Pareto [YAN02].

A base des solutions optimales retenues au sens de Pareto, nous définissons une nouvelle fonction objectif F unique qui fait le compromis entre les deux objectifs : $\sum \text{coût}$ et $\sum \text{distance}$. La manière la plus simple de procéder consiste à prendre chacune des fonctions objectif, à leur appliquer un coefficient de pondération et à faire la somme des fonctions objectif pondérées. On obtient alors une nouvelle fonction objectif F comme étant la somme pondérée des différentes fonctions objectifs. En affectant un coefficient de poids qui représente l'importance relative que nous attribuons à l'objectif :

$$F = \alpha * \sum \text{coût} + \beta * \sum \text{distance} \text{ où les poids } \alpha \text{ et } \beta \in [0..1] \text{ avec } \alpha + \beta = 1.$$

La définition de la fonction F , nous permettra d'en déduire, la trajectoire poursuivie par le data ferry dans les trois configurations : (1) $\alpha = \beta = 0.5$: les deux objectifs sont de même importance, (2) $\alpha > \beta$: le premier objectif est favorisé par rapport au deuxième objectif, (3) $\alpha < \beta$: le deuxième objectif est favorisé par rapport au premier objectif.

3.6. Conclusion

Dans ce chapitre, nous avons détaillé notre solution PSFPP pour la planification de la trajectoire de data ferry, qui incorpore un nouveau modèle de connectivité probabiliste et de prédictions des coûts de communication.

Le problème de la planification que nous avons présenté est formalisé par un problème d'optimisation multi-objectif. La résolution de ce problème est effectuée en étapes. Dans chaque étape, nous avons présenté les méthodes et les concepts à utiliser pour la résolution. Premièrement, nous avons défini une approche de prédiction des coûts de communication pour résoudre le problème de connectivité et la durée de vie du réseau. Deuxièmement, nous avons formulé notre problème de planification qui incorpore le modèle de connectivité établi. Troisièmement, nous avons proposé une solution à ce problème de planification dont la résolution est effectuée en plusieurs étapes.

La solution PSFPP de la planification détaillée dans ce chapitre constitue une solution pour un environnement réel et elle incorpore un modèle de communication réaliste. L'environnement ciblé par cette solution peut être représenté par un champ de bataille composé d'un ensemble de capteurs et un drone pour la collecte de données.

Chapitre 4 Implémentation et évaluation de la solution PSFPP proposée

4.1. Introduction

Ce chapitre est consacré à l'implémentation et l'évaluation de la solution PSFPP proposée pour la planification de la trajectoire de data ferry. Des scénarios de test sont implémentés pour évaluer les caractéristiques de la solution PSFPP. Plusieurs parties sont incluses dans ce chapitre. Le chapitre commence par décrire les heuristiques TSP et SSP utilisées pour la résolution de notre problème d'optimisation. Ensuite, une étude expérimentale est effectuée pour la validation de modèle de communication probabiliste proposé. Une implémentation de la solution PSFPP est réalisée à l'aide de l'environnement de développement Visual C++ afin de décrire la trajectoire poursuivie par le data ferry. Dans la dernière partie une simulation sous OMNET++ est proposée pour évaluer les performances de la solution PSFPP et effectuer une comparaison qualitative de la solution proposée qui adopte un modèle de communication probabiliste reflétant mieux la réalité à la planification classique TSPN (qui modélise le champ de communication par forme géométrique abstraite).

Le but de ce chapitre est d'évaluer les performances de notre solution et de décrire les différentes configurations (latence, énergie) possibles pour un environnement visé.

4.2. Les heuristiques TSP et SSP

Dans cette partie, nous allons décrire les heuristiques TSP et SSP utilisées pour l'implémentation de notre solution. L'heuristique la mieux adaptée à notre solution en fonction de temps d'exécution et l'optimalité des solutions est choisie.

4.2.1. Le problème de voyageur de commerce TSP

Le problème TSP est l'un des plus vieux problèmes combinatoires. Des mathématiciens s'y sont intéressés depuis le début du vingtième siècle cherchant à apporter une réponse à ce problème. La définition en est simple : soit un ensemble de nœuds et d'arêtes munies de coûts. Le problème consiste à trouver un circuit Hamiltonien, c'est-à-dire le circuit passant par tous les nœuds une et une seule fois et qui minimise le coût total lié aux arcs empruntés par le voyageur.

Une formulation a été proposée par (Dantzig et al. 1954). Pour les nœuds i et j , c_{ij} représente le coût du trajet entre ces deux nœuds, et X_{ij} une variable binaire indiquant si le trajet est compris dans la solution. Le TSP se modélise alors comme suit :

$$\text{Minimiser } Z = \sum_{i \in N} \sum_{j \in N} X_{ij} C_{ij} \quad (\text{IV.1})$$

Sous les contraintes :

$$\sum_{i \in N} X_{ij} = 1 \quad \forall j \in N \quad (\text{IV.2})$$

$$\sum_{j \in N} X_{ij} = 1 \quad \forall i \in N \quad (\text{IV.3})$$

$$\sum_{i,j \in S} X_{ij} \leq |S| - 1 \quad \forall S \subset N \text{ telle que } 2 \leq |S| \leq N - 2 \quad (\text{IV.4})$$

Les contraintes (2) et (3) assurent que le voyageur entre et sort une seule fois de chaque nœud (Conservation des flots). La contrainte (4) est une formulation classique pour éviter les sous-tours.

Nous allons choisir les heuristiques de complexité polynomiale de l'ordre $O(n^2)$ (n : nombre des nœuds) qui traitent le problème TSP dans un temps réduit. Parmi lesquelles nous citons :

a) La méthode d'insertion

Initialement composé d'un cycle de trois nœuds, l'algorithme d'insertion a pour but de construire un TSP pas à pas. L'étape suivante consiste à insérer un nouveau nœud dans le cycle de manière optimale en trouvant la position qui augmente le moins la longueur totale du cycle déjà construit. L'algorithme s'arrête lorsque tous les nœuds à visiter sont insérés.

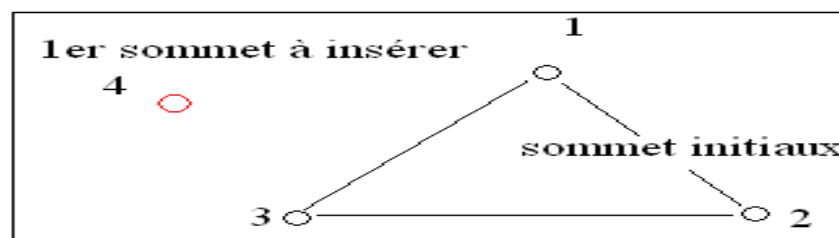


Figure 4.1. Fonctionnement de l'algorithme d'insertion.

b) La méthode de plus proche insertion

Cette approche est basée sur l'approche précédente, l'insertion d'un nouveau nœud dans le cycle se fait en comparant la distance entre le nœud à insérer avec chaque arrête du chemin construit, l'arrête avec la distance minimale est sélectionnée, après le nouveau nœud est alors inséré entre les deux nœuds reliés par cette arrête comme le montre la figure ci-dessous.

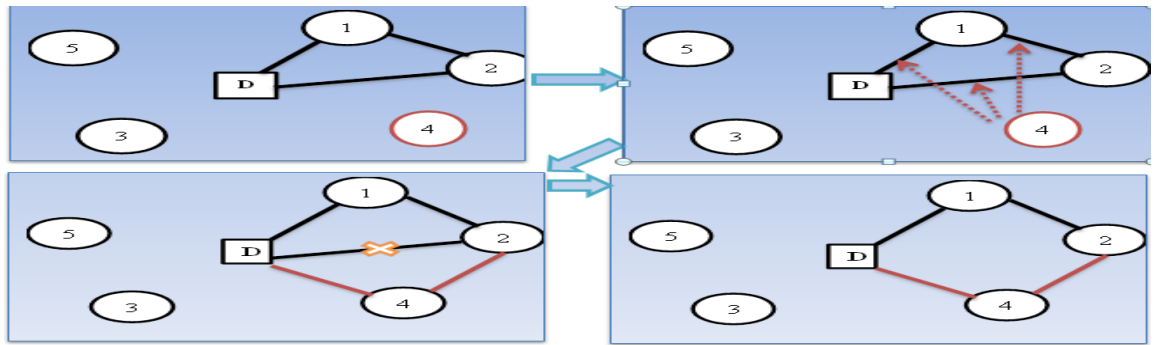


Figure 4.2. Fonctionnement de l'algorithme de la plus proche insertion.

c) La méthode de plus proche voisin

C'est un algorithme qui construit la solution pas à pas en démarrant d'un point et à chaque fois le plus proche voisin du sommet ajouté est inséré. Le principe du plus proche voisin est basé sur la sélection du prochain nœud tel que le coût entre le nœud courant et le prochain soit minimal.

d) Le recuit simulé

C'est une heuristique qui cherche la solution optimale à partir d'une solution existante. Il est utilisé pour améliorer la performance des résultats (recherche de la solution optimale).

Notre choix se focalise sur l'algorithme de plus proche voisin parce qu'il s'agit d'une méthode déterministe la plus simple pour évaluer une seule fois le chemin entre un ensemble de nœuds. L'algorithme de recuit simulé est appliqué sur le chemin résultant afin d'améliorer la solution.

4.2.2. Le SSP (Sub Set Sum Problem)

Le SSP est un problème NP-complet. La formulation générale de problème est : étant donné un ensemble d'éléments n et une variable k avec :

$$\begin{cases} w_j = \text{le poid de l'élément } j \\ c = \text{la valeur de la variable } K \end{cases}$$

Le problème est de sélectionner un sous-ensemble d'éléments dont le poids total doit être égale à la valeur c . Le SSP se modélise alors comme suit :

$$\text{Maximiser } z = \sum_{j=1}^n w_j x_j$$

Sous les contraintes :

$$\sum_{j=1}^n w_j x_j = c$$

$$x_j = 0 \vee 1, j \in N = \{1, \dots, n\}$$

$$\text{avec : } x_j = \begin{cases} 1 & \text{si l'élément } j \text{ est sélectionné} \\ 0 & \text{sinon} \end{cases}$$

Le SSP est résolu par programmation dynamique. En entrée : on a un vecteur $A [1 \dots n]$ des valeurs d'une colonne j de la matrice MCGR codées en décimale et une valeur entière $K=15$.

Le problème est résolu comme suit :

Soit $S[i,j]$ une matrice des valeurs booléennes définies comme vrai s'il y a un sous-ensemble des éléments $A[1 \dots i]$ dont la somme égale à j . $S[n,K]$ indique s'il y a des sous-ensembles de couverture. En général :

$$S[i,j] = S[i-1, j-A[i]] \vee S[i-1,j]$$

Les conditions initiales sont $S[i,0]=\text{vrai}$, et $S[0,j]=\text{faux}$, pour $j > 0$.

L'algorithme est de complexité $O(n \cdot K)$, il permet la détermination des sous-ensembles de couverture par des retours arrière.

4.3. Etude expérimentale du modèle de communication probabiliste

Dans cette section, nous évaluons par des mesures, en utilisant le XBee Série 1 (IEEE 802.15.4), le RSSI (Received Signal Strength Indicator) dans un environnement intérieur afin d'examiner les caractéristiques de base d'un canal de communication IEEE 802.15.4. Cette étude nous permet de calibrer correctement notre modèle d'estimation de coût de communication.

La première étape dans la validation du modèle de coût de communication proposé est de vérifier le modèle path loss dans l'environnement du laboratoire (Labo I.A à l'EMP). La figure.4.3 montre la moyenne de puissance reçue mesurée et la puissance reçue théorique décrite par l'équation III.2 avec $\gamma = 3.2$, $\sigma_e = 4$, $PL(d_0) = 40\text{dB}$, $d_0 = 1$ et une puissance de sortie $P_t = 0\text{dBm}$. D'après les résultats obtenus, nous pouvons voir que les données expérimentales correspondent bien au modèle path loss [RAP01]. Ces valeurs seront utilisées dans les prochaines simulations.

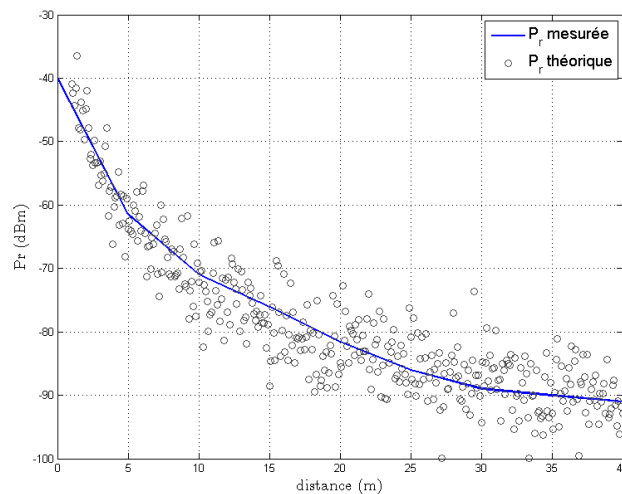


Figure 4.3. Comparaison entre la puissance reçue mesurée et la puissance reçue théorique.

4.4. Evaluation de la trajectoire poursuivie par le data ferry

Une description et analyse de la trajectoire poursuivie par le data ferry dans une scène composée de nœuds capteurs déployés aléatoirement est effectuée pour les différents environnements : (i) environnement favorisant l'optimisation d'énergie ($\alpha > \beta$), (ii) environnement favorisant la minimisation de la latence ($\alpha < \beta$) et (iii) environnement qui fait le compromis entre l'énergie et la latence ($\alpha = \beta$). Nous avons envisagé une implémentation de la solution à l'aide de l'environnement de développement Visuel C++. Ensuite, une description schématique de la trajectoire poursuivie par le data ferry dans les trois configurations : $\alpha > \beta$, $\alpha < \beta$ et $\alpha = \beta$.

4.4.1. Implémentation

Pour l'implémentation des différents algorithmes, nous avons utilisé l'IDE Microsoft Visuel C++ 2010 comme langage de programmation orienté objet. Nous avons choisi l'environnement orienté objet pour faciliter le développement, la maintenance et la réutilisation des différents modules que nous avons implémenté.

4.4.2. Etude de la trajectoire de data ferry

Pour décrire et évaluer la trajectoire poursuivie par le data ferry dans la solution proposée, nous allons envisager trois scénarios, nous commençons d'abord par varier les paramètres α et β de la fonction objective $F = \alpha * \sum \text{coût} + \beta * \sum \text{distance}$ afin d'observer la variation de la trajectoire pour les différents environnements et effectuer une comparaison. Trois cas de figures se présentent :

- 1) Favoriser l'optimisation de l'énergie : $\alpha > \beta$.
- 2) Favoriser la latence de la collecte de données : $\alpha < \beta$.
- 3) L'optimisation de l'énergie et la latence sont de même pondération : $\alpha = \beta = 1/2$.

Dans les trois scénarios, nous sélectionnons le point optimal de l'ensemble des solutions réalisables, nous réalisons le tracé de la trajectoire. Ensuite, nous commentons les trajectoires obtenues dans les trois configurations : La scène est de dimension 15x15 cellules, la densité des nœuds est fixée à 16 nœuds et la distance seuil $d_s = 2$ cellules.

Nous avons effectué une analyse quantitative de la trajectoire poursuivie par le data ferry dans les trois cas de figures.

-Favoriser l'optimisation de l'énergie : $\alpha > \beta$:

La figure 4.4 montre la trajectoire poursuivie par le data ferry dans le cas où l'optimisation de l'énergie est favorisée par rapport à la latence. Nous constatons que la trajectoire du data ferry se rapproche le plus possible des capteurs. Ceci s'explique par le fait de la minimisation de l'énergie consommée qui passe nécessairement par la minimisation de la distance qui sépare le data ferry du capteur. Dans ce cas le data ferry a tendance à se rapprocher au mieux des capteurs déployés.

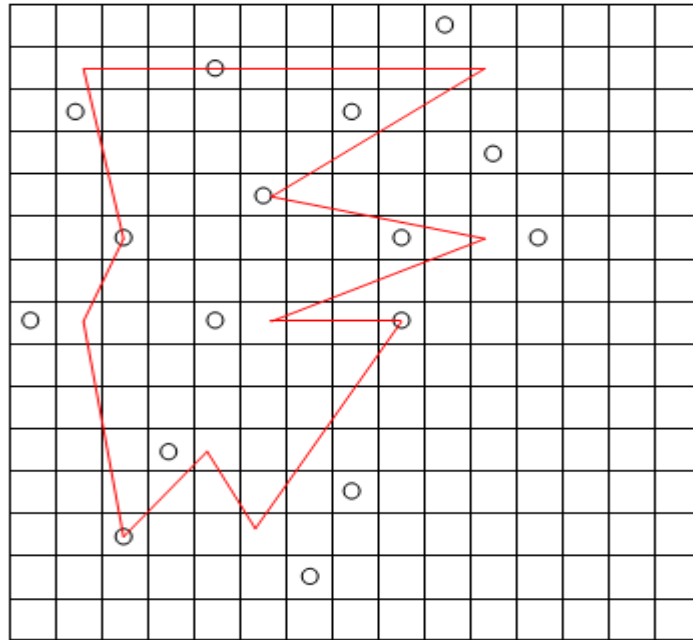


Figure 4.4. La trajectoire du data ferry avec $\alpha > \beta$.

-Favoriser la latence de la collecte de données : $\alpha < \beta$:

La figure 4.5 montre la trajectoire poursuivie par le data ferry dans le cas où la minimisation de la latence est favorisée par rapport à la consommation d'énergie. Puisque le data ferry se déplace à une vitesse constante, la minimisation de la latence est équivalente à la minimisation de la distance totale parcourue par le data ferry. Sur la figure 4.5, nous constatons que la distance entre le data ferry et les capteurs est arbitraire. En effet, le data ferry s'intéresse peu à la distance qui le sépare des capteurs, et suit simplement la trajectoire qui lui permet de minimiser la distance totale parcourue afin de minimiser la latence.

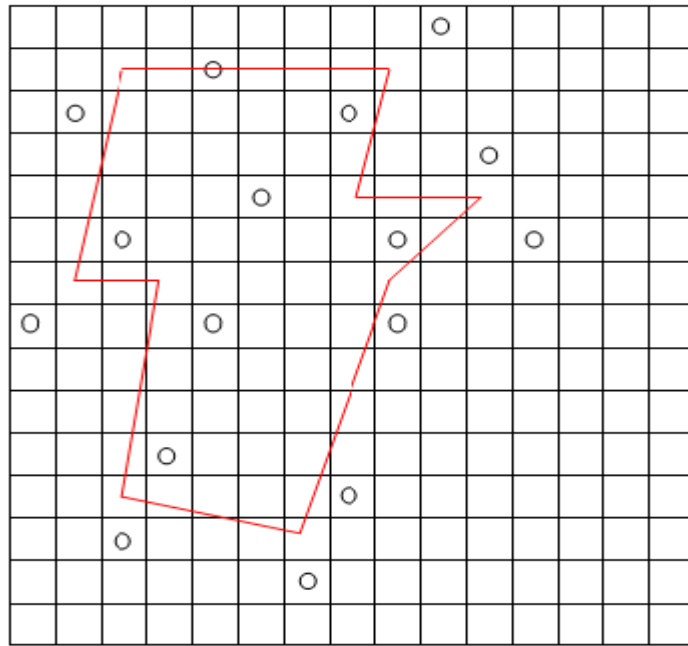


Figure 4.5. La trajectoire du data ferry avec $\alpha < \beta$.

-L'optimisation de l'énergie et la latence sont de même pondération : $\alpha = \beta$:

La figure 4.6 montre la trajectoire poursuivie par le data ferry dans le cas où l'optimisation de l'énergie et la latence sont de même pondération. Dans ce cas, le data ferry tente de minimiser à la fois les distances qui le séparent des capteurs ainsi que la distance totale parcourue. Sur la figure 4.6, nous constatons que la distance entre le data ferry et les capteurs est pratiquement uniforme, ceci est justifié par le compromis entre la latence et l'énergie.

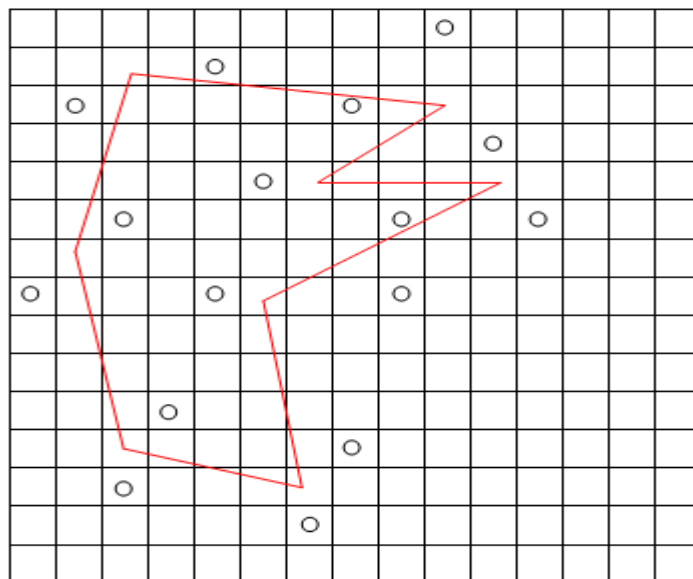


Figure 4.6. La trajectoire du data ferry avec $\alpha = \beta$.

4.5. Evaluation de la solution proposée

La simulation s'agit d'implanter un modèle simplifié du système à l'aide d'un programme de simulation adéquat. Cette méthode traduit le comportement du système à évaluer d'une manière réaliste. La simulation permet en plus de visualiser les résultats sous forme de graphes faciles à analyser et à interpréter. La simulation n'est pas une méthode exacte, et nécessite de prêter une attention particulière aux interprétations des résultats. La simulation a pour avantage de permettre l'étude du système en variant ses divers paramètres.

Dans cette partie, nous avons mesuré à l'aide de la simulation l'impact de la variation de la distance seuil d_s sur les deux caractéristiques : énergie et latence de la solution PSFPP proposée. Ensuite, nous avons envisagé une comparaison entre la solution PSFPP et la planification classique TSPN pour mesurer l'impact de la variation de la densité des nœuds sur les deux caractéristiques : énergie et latence.

Les scénarios de simulation choisis sont de deux types. Le premier scénario consiste à évaluer uniquement les caractéristiques de la solution proposée PSFPP en fonction de la variation de la distance seuil d_s . Le deuxième scénario consiste à effectuer une comparaison entre la solution PSFPP proposée et la planification classique TSPN.

4.5.1. Environnement de simulation

OMNeT++ est une application développée par Andras Varga, chercheur à l'université de Budapest. Il possède une architecture générique de sorte qu'il peut être utilisé pour la modélisation des réseaux câblés, les communications sans fil, la modélisation de protocoles, la modélisation des multiprocesseurs des systèmes distribués, l'évaluation des aspects de la performance des systèmes logiciels complexes, la validation d'architectures matérielles,...etc.

Il est totalement paramétrable programmable et modulaire. Il fournit des outils, des modèles et des infrastructures qui aident à réaliser une simulation théorique. Les modèles sont assemblés à partir des composants réutilisables appelés modules. Ces modules peuvent être combinés de différentes manières.

Les simulations par OMNeT ++ peuvent être exécutées sous différentes interfaces d'utilisateur : interfaces graphiques, interfaces d'animation et les interfaces utilisateurs de commande en ligne qui sont les meilleurs. De plus, il est disponible pour les systèmes

d'exploitation les plus courants (Linux, Mac OS / X, Windows).

La mise en œuvre d'une simulation se fait via une étape qui décrit la topologie du réseau (la mobilité) et le comportement de ses composants (la trafic), puis vient l'étape de simulation proprement dite et enfin l'extraction et la visualisation de ces résultats.

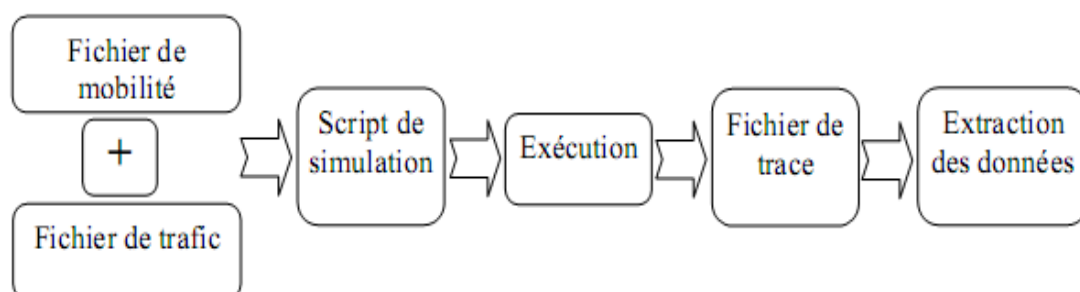


Figure 4.7 Processus de simulation.

4.5.2. Paramètres de simulation

Pour le modèle de mobilité de data ferry, nous avons utilisé les classes qui nous permettent de créer un modèle de mobilité spécifié à notre solution. La classe abstraite de base *Imobility* définissant l'interface publique qui doit être fournie par tous les modules de mobilité. La classe *LinearMobility* qui définit un modèle linéaire du mouvement. L'algorithme SCD («Stop to Collect Data») est utilisé pour ajuster la vitesse du data ferry, le principe est que le data ferry se déplace à une vitesse constante v et s'arrête pour un temps Δt aux capteurs ayant des données à collecter.

Dans la solution PSFPP, Le data ferry se déplace dans la scène selon des coordonnées (x_i, y_i) qui représentent la liste des positions de mobile. Ces coordonnées sont stockées dans un tableau *ferry_position*. Lorsque le data ferry est dans une position (x_i, y_i) , il émet un message « hello » à l'ensemble des capteurs couverts par cette position. Un nœud capteur qui détecte le message « hello » transfère ses données stockées de son buffer vers le *ferry_buffer*.

Dans la planification TSPN, les nœuds capteurs sont positionnés dans les cellules. Le rayon de communication d'un capteur = taille de la cellule/2. Avec cette considération, il n'y a pas une communication entre les nœuds capteur. L'application d'une heuristique TSPN permet la détermination de la liste des positions de mobile. Le data ferry dans une position (x_i, y_i) collecte les données d'un seul capteur.

L'ensemble des paramètres utilisés dans la simulation effectuée dans ce mémoire sont résumés dans le tableau 4.1 :

le temps d'attente pour la collecte de données (Δt)	1s
la vitesse de déplacement de data ferry (v)	10 m/s
Couche MAC	IEEE 802.15.4
Portée de transmission d'un capteur pour TSPN	20 m
Bande passante	200 Kb/s
Débit d'envoi des données	8 paquets/s

Tableau 4.1. Les paramètres utilisés pour la simulation.

4.5.3. Paramètres d'étude

Pour l'évaluation et la comparaison de la solution PSFPP proposée nous pouvons varier les paramètres suivants :

- Distance seuil d_s : la distance introduite pour éviter les liens de communication de mauvaise qualité. Ce paramètre est utilisé pour l'évaluation des caractéristiques de la solution proposée.
- Densité des nœuds : nombre de nœuds déployés dans la région d'intérêt par rapport à la surface de cette dernière. Ce paramètre est utilisé pour la comparaison entre la solution proposée et la planification TSPN.

4.5.4. Métrique à mesurer

Deux métriques qui permettent d'évaluer les caractéristiques de notre solution :

- (1) La latence de la collecte de données : une métrique relative aux utilisateurs.
- (2) La consommation d'énergie : une métrique relative à l'utilisation des ressources.

Pour évaluer les caractéristiques de notre solution nous avons mesuré les métriques suivantes :

- Energie (erc) : cette métrique indique l'énergie totale du réseau consommée. Formellement, elle est donnée par : $erc = \sum_{k=1}^{nr} ec_k$, où ec_k désigne l'énergie consommée par le nœud k et nr le nombre de nœuds formant le RCSF.
- Latence (lcd) : cette métrique indique la latence de la collecte de données par le data ferry pour une trajectoire donnée. Formellement, elle est donnée par : $lcd = \text{longueur de la trajectoire} / v + \Delta t * Nb_{\text{waitpoint}}$

où v désigne la vitesse de data ferry et $Nb_{waitpoint}$ le nombre total des positions dans lesquelles le data ferry s'arrête pour la collecte de données.

4.5.5. Scénarios de simulation

Pour essayer d'approcher au mieux la réalité, nous avons choisi des scénarios de simulation adéquats. Pour ce faire, les scénarios choisis ont distribution aléatoire de nœuds. Dans les deux scénarios, nous avons choisi une configuration où les deux paramètres : énergie et latence sont de même importance ($\alpha=\beta=0.5$).

4.5.5.1. Scénario 1 : Variation de la distance seuil d_s

Ce scénario est utilisé pour mesurer l'impact de la variation de la distance seuil d_s sur les deux paramètres de simulation : énergie (erc) et latence (lcd) à évaluer pour notre solution. Nous faisons varier la distance seuil d_{seuil} entre 10m et 30m. Nous fixons le nombre des nœuds à 200, la dimension de la scène à 50x50 cellules. La taille de la cellule est fixée à 10m.

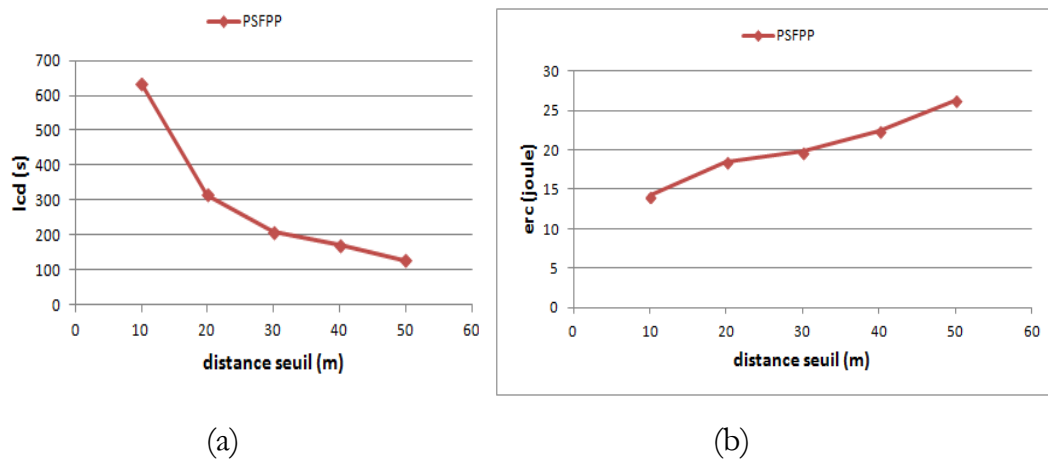


Figure 4.8 Impact de la distance seuil sur les caractéristiques de la solution

La figure 4.8 montre l'impact de la variation de la distance seuil sur la latence et la consommation d'énergie. Selon les résultats obtenus, on remarque que l'augmentation de la distance seuil génère une augmentation dans l'énergie consommée (figure 4.8 (b)) et une diminution dans la latence de la collecte de données (figure 4.8 (a)).

4.5.5.2. Scénario 2 : Variation de la densité des nœuds

Ce scénario est utilisé pour la comparaison entre la solution proposée et la planification classique TSPN pour mesurer l'impact de la variation de la densité des nœuds sur les deux caractéristiques : énergie (erc) et latence (lcd). La taille de la cellule est fixée à 40m. La distance seuil d_s à 80m. Le rayon de communication d'un nœud pour la planification

TSPN à 20m. La dimension de la scène à 50x50 cellules. Nombre de nœuds dans l'ensemble {16, 32, 64, 128, 256, 512}.

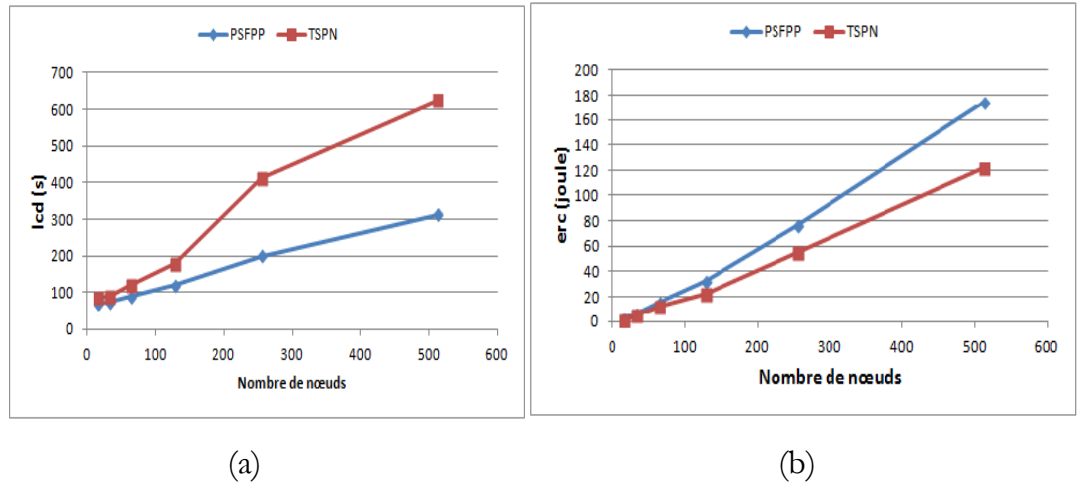


Figure 4.9. Impact de la variation de la densité sur les caractéristiques de la solution.

La figure 4.9 montre que les solutions PSFPP et TSPN ont des résultats comparables pour la latence et la consommation d'énergie dans des scènes de faible densité (inférieur à 100 nœuds). Cependant, lorsque la densité augmente notre solution donne un résultat meilleur pour la latence que la solution TSPN (figure 4.9 (b)). La solution TSPN donne des résultats meilleurs pour la consommation d'énergie (figure 4.9 (b)). Avec l'augmentation de nombre de nœuds il y'a une augmentation dans la latence et l'énergie pour les deux solutions. Dans le cas d'un réseau d'une petite échelle, les deux solutions ont des résultats comparables. Dans le cas d'un réseau d'une grande échelle, les deux solutions ont des résultats différents la latence est meilleur dans PSFPP et la consommation d'énergie est meilleur dans TSPN.

4.5.6. Synthèse des résultats de simulation

L'augmentation de la distance seuil ds a un impact positif dans la solution PSFPP pour la latence et un impact négatif pour l'optimisation de l'énergie (figure 4.8). L'explication est basée sur le fait que l'augmentation de la distance seuil permet de couvrir un maximum de capteurs, par conséquent, une diminution dans la distance parcourue par le data ferry. En revanche, cette augmentation de la distance seuil génère un nombre important de paquets perdus ce qui nécessite une retransmission des paquets et par conséquent, une augmentation dans l'énergie consommée.

L'explication donnée aux résultats de la comparaison (figure 4.9). Dans la planification TSPN, le data ferry doit assurer la collecte de données de chaque capteur individuellement, par conséquent, ceci engendre une fiabilité et une optimisation de l'énergie aux niveaux des capteurs mais avec une augmentation dans la latence de la collecte de données. En contrepartie, un phénomène inverse est observé pour la solution PSFPP ceci due au fait que le data ferry dans cette solution collecte les données à partir des capteurs sans être à proximité des nœuds capteurs.

4.6. Conclusion

Dans ce chapitre, nous avons implémenté et évalué, à l'aide de l'environnement de développement Visual C++ et le simulateur OMNET++, la solution PSFPP proposée pour la planification de la trajectoire de data ferry avec prise en compte de modèle de communication probabiliste. Nous avons commencé par effectuer une étude expérimentale pour la validation de modèle de communication proposé. Puis, nous avons effectué une étude sur la trajectoire poursuivie par le data ferry dans les trois cas de configuration. Enfin, nous avons simulé la solution PSFPP proposée en effectuant une comparaison avec la planification classique TSPN.

Les résultats obtenus, par l'implémentation et la simulation, ont montré que la solution proposée peut être adoptée selon les besoins de l'environnement visé par ajustement des paramètres α et β de la fonction objective.

Conclusion générale et perspectives

Dans ce mémoire, nous avons proposé une solution pour la planification de la trajectoire du data ferry basée sur un modèle de communication probabiliste pour la collecte de données dans un environnement réel. D'après notre recherche bibliographique, aucune proposition similaire n'existe dans le domaine. En effet, les approches existantes assument un modèle de communication binaire alors que la notre adopte un modèle de communication probabiliste qui reflète encore mieux la réalité.

Nous avons présenté un état de l'art des approches de la planification de la trajectoire des data ferries, nous avons proposé une nouvelle classification pour la planification de la trajectoire classe TSP tout en effectuant une synthèse des travaux existants.

Après avoir proposé un nouveau modèle de prédictions des coûts de communication, nous avons formulé le problème de planification de la trajectoire du data ferry sous forme d'un problème d'optimisation multi-objectif. Pour résoudre ce problème, nous avons procédé par étapes en exploitant des heuristiques du type TSP et SSP. Nous avons proposé une nouvelle heuristique pour la détermination des sous-ensembles de couverture de la scène basée sur le problème SSP et nous avons appliqué une heuristique TSP pour la détermination des distances minimales de ces sous-ensembles de couverture.

Nos propositions ont été validées à la fois par des simulations et des tests expérimentaux.

Sur la base du travail réalisé, nous dressons des perspectives de recherche et de développement qui sont les suivants :

- La révision de la modélisation du problème, en utilisant par exemple un échantillonnage aléatoire au lieu de la discrétisation. Cela permettra de réduire la complexité du problème ;
- Définir des mécanismes pour la détermination de la distance seuil d_{seuil} en fonction des paramètres et caractéristiques de data ferry.
- L'intégration de la dynamique de data ferry dans la proposition ;
- L'exploitation de la communication entre capteurs voisins et le concept des points de rendez-vous pour réduire d'avantage la complexité du problème dans le cas des grandes instances.
- La validation de nos propositions par le déploiement d'un banc d'essai de RCSF avec collecteur mobile.

Références bibliographiques

- [AKY02]. Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. Wireless Sensor Networks : a Survey. Computer Networks, 2002.
- [AMU09] Amundson I, Xenofon D and Koutsoukos. A Survey on Localization for Mobile Wireless Sensor Networks, MELT, LNCS, Springer, USA, 2009.
- [ANA07] Anastasi G, Conti M, Monaldi E, Passarella A. An Adaptive Data-transfer Protocol for Sensor Networks with Data Mules. IEEE , 2007.
- [ANA08] Anastasi G, Conti M, Francesco D. Data Collection in Sensor Networks with Data Mules: an Integrated Simulation Analysis. IEEE, 2008.
- [BEY09] BEYDOUN Kamel, Conception d'un protocole de routage hiérarchique pour les réseaux de capteurs , Thèse, Université de Franche-Comte, 2009.
- [BOR08] Borges J, Andrade G. Unmanned Vehicles for Environmental Data Collection. The NATO/CCMS Pilot Project, Springer, 2008.
- [BOR10] Borsetti D, Casetti C, Fabiana C. Virtual Data Mules for Data Collection in Road-Side Sensor Network, ACM, 2010.
- [BOU07] BOUABDELLAH Kechar, Problématique de la consommation de l'énergie dans les réseaux de capteurs sans fil, Séminaire LIUPPA, Université de Pau et des Pays de l'Adour, 14 Octobre 2007.
- [CHA03] Chakrabarti A, Sabharwal A, Aazhang B. Using Predictable Observer Mobility for Power Efficient Design of Sensor Networks. Proceedings of the 2nd International Workshop on Information Processing in Sensor Networks, 2003.
- [CHO03] Chong C.Y, Kumar S.P. Sensor Networks: Evolution, Opportunities, and Challenges. Proceedings of the IEEE, 2003.
- [CRE03] Creuze V, Jouvencel B. Perception et suivi de fond pour véhicules autonomes sous-marins. Rapport de recherche, Université Montpellier 2, 2003.
- [DEE11] Deepak B., Onur T. and Volcan Isler. Robotic Data Mules for Collecting Data over Sparse Sensor Fields. Journal of field robotics Wiley periodicals. 2011.
- [DEL09] Delia Ciullo, Guner D.Celik, Eytan Modiano. Minimizing Transmission Energy in Sensor Networks via Trajectory Control. Mitor Project a research collaboration. 2009.
- [DER11] DERAFA laloui . Contribution à la mise en œuvre d'une commande référencée vision : application à une plateforme volante autopilotée, Thèse doctorat, école militaire polytechnique, 2011.
- [EKI06]. Ekici, E., Gu, Y., and Bozdag, D. Mobility-based Communication in Wireless Sensor Networks. IEEE Communications Magazine. 2006.

- [ERM08] Erman T, Hoesel L, Havinga P. Enabling Mobility in Heterogeneous Wireless Sensor Networks Cooperating With UAVs for Mission-Critical Management, IEEE, 2008.
- [FAN05] Fang, L., Antsaklis, P.J., Montestruque, L., McMickell, M.B., Lemmon, M., Sun, Y., Fang, H., Koutroulis, I., Haenggi, M., Xie, M., Xie, X.: Design of a Wireless Assisted Pedestrian Dead Reckoning System – the NavMote experience. IEEE Transactions on Instrumentation and Measurement, 2005
- [FAN10] Fang , and Y.Tseng. Energy-Conserving Data Gathering by Mobile Mules in a Spatially Separated Wireless Sensor Network.Wireless communications and mobile computing. 2010.
- [FIN10] Finn A, Scheduling S. Developments and Challenges for Autonomous Unmanned Vehicles. Intelligent Systems Reference Library, Springer, 2010.
- [GAN03].Gandham S-R, Dawande M, Prakash R, Venkatesan S. Energy Efficient Schemes for Wireless Sensor Networks with Multiple Mobile Base Stations, IEEE, GLOBECOM Conference, 2003.
- [GER94] Gerhard R. The Traveling Salesman, Computational Solutions for TSP Applications, Springer, 1994.
- [GER09]. Gérard Chalhoub : Réseaux de capteurs sans fils. Rapport de recherche : Clermont Université, 2009.
- [GIL92] Gilbert L. The traveling salesman problem: An overview of exact and approximate algorithms. European Journal of Operational Research, 1992.
- [HAR09] Harold C, Ting H, Kang-won L, Kin K. Dynamic Control for Data Ferries Under Partial Observations, Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defense, 2009.
- [HEI05] Heidemann J., Yuan L., Affan S. Underwater Sensor Networking: Research Challenges and Potential Applications. USC/ISI Technical report, 2005.
- [HIL03] Hill J.L. System Architecture for Wireless Sensor Networks.University of California, Berkeley, PhD Thesis, 2003.
- [KAN04] Kansal A, Somasundara AA, Jea DD, Srivastava MB, Estrin D. Intelligent Fluid Infrastructure for Embedded Networks. Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services (MobiSys), 2004.
- [KHE04] Khelladi L, Badache N. Les réseaux de capteurs : état de l'art. Rapport de recherche, Algérie, Février 2004.
- [KOU11] Kouah R et Moussaoui S. Mobilité des nœuds dans les réseaux de capteurs sans fil, In CGE'07 EMP, ALGERIE, 12-13 Avril, 2011.

- [KUM13] Kumar V.'Ads' Algorithm for Subset Sum Problem, International journal of computer applications, march 2013.
- [LIU05] Liu B, Brass P, Dousse O, Nain P. Mobility Improves Coverage of Sensor Networks. Proceedings of the 6th ACM international symposium on Mobile ad-hoc networking and computing, 2005.
- [MAR11] Mario D, SAJAL K, Crewman D, Anastasi G. Data Collection in Wireless Sensor Networks with Mobile Elements: A Survey, ACM, 2011.
- [MAT06] Matthew D., Peter C. Vasilescu I. Daniela R. Data Muling over Underwater Wireless Sensor Networks using an Autonomous Underwater Vehicle. Proceedings of the IEEE international conference on robotics and automation, 2006.
- [MAY07] Ma M, Yang Y. SenCar: an Energy-efficient Data Gathering Mechanism for Large-scale Multi-hop Sensor Networks IEEE Trans. Parallel and Distributed Systems, 2007.
- [MAY08] M. Ma and Y. Yang. Data gathering in Wireless Sensor Networks with Mobile Collectors IEEE Int Parallel and Distributed Processing Symp, 2008.
- [MUN07] Munir, S.A., Ren, B., Jiao, W., Wang, B., Xie, D., Ma, J. Mobile Wireless Sensor Network: Architecture and Enabling Technologies for Ubiquitous Computing. International Conference on Advanced Information Networking and Applications Workshops, 2007.
- [NAT11] Natalizio E, Loscri V. Controlled Mobility in Mobile Sensor Networks: Advantages, issues, and Challenges. Springer, 2011.
- [NES06] Nesamony S, Vairamuthu M, Orlowska M, Sadiq S. On Optimal Route Computation of Mobile Sink in a Wireless Sensor Network. Technical Report 465. ITEE, 2006.
- [NES07] Nesamony S, Vairamuthu MK, Orlowska ME. On Optimal Route of a Calibrating Mobile Sink in a Wireless Sensor Network. Proceedings of the 4th International Conference on Networked Sensing Systems, 2007.
- [NIC76] Nicos C. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical Report 388, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, 1976.
- [RAO08] Rao J, Biswas S. Joint Routing and Navigation Protocols for Data Harvesting in Sensor Networks. Proceedings of the 5th IEEE International Conference on Mobile Ad-hoc and Sensor Systems. 2008.
- [RAP01] Rappaport T. Wireless Communications: Principles and Practice. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 2001.
- [REN06] Ren B, Ma J, Chen C. The Hybrid Mobile Wireless Sensor Networks for Data Gathering. IWCMC'06, Vancouver, British Columbia, Canada, July 3–6, 2006.

- [ROD99] Rodoplu V, Meng TH. Minimum Energy Mobile Wireless Networks. IEEE J Sel Areas Commun 1999.
- [SHA03]. Shah, R. C., Roy, S., Jain, S., and Brunette, W. Data mules: Modeling a three-tier Architecture for Sparse Sensor Networks. In Proceedings of the 2nd ACM International Workshop on Wireless Sensor Networks and Applications , 2003.
- [SOM06]. Somasundara, A., Kansal, A., Jea, D., Estrin, D., and Srivastava, M. Controllably mobile infrastructure for low energy embedded networks. IEEE Transactions on Mobile Computing , 2006.
- [SUG07] Sugihara R, Gupta RK. Scheduling under location and time constraints for data collection in sensor networks”. Proceedings of the 28th IEEE Real-Time Systems Symposium (RTSS) Work in Progress Session, 2007.
- [SUG08] Sugihara R, Gupta RK. Improving the data delivery latency in sensor networks with controlled mobility. Proceedings of the 4th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS), 2008.
- [SEV06] Severine S, "Architecture logicielle pour capteurs sans-fil en réseau", Master TI 2e année, Mälardalen University, Sweden, Janvier-Juin 2006.
- [TIN03] TinyOS. www.tinyos.net/tinyos-1.x/doc/, 2003.
- [TOD06] Todd M., Mascarenas D., Flynn E. and Rosing T. A different approach to sensor networking for SHM: Remote Powering and interrogation with unmanned aerial vehicles, IEEE, 2006.
- [VAS05] Vasilescu, Kotay K. and Rus D. Data collection, storage, and retrieval with an underwater sensor network.ACM, 2005.
- [WAN09] Wang Y,Wu J, and Tseng Y, Mobility Management Algorithms and Applications for Mobile Sensor Networks ,Wiley wireless communications and mobile computing, 2009.
- [XING07] Xing G, Wang T, Xie Z, Jia W. Rendezvous planning in mobility-assisted wireless sensor networks. Proceedings of the 28th IEEE International Real-Time Systems Symposium (RTSS), 2007.
- [XING08] Xing G, Wang T, Jia W, Li M. Rendezvous design algorithms for wireless sensor networks with a mobile base station. Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), 2008.
- [YAN02] Yann C, Patrick S, optimisation multi-objectif, édition EYROLLES, 2002, ISBN 2-212-11168-1.
- [YAO05] Gu, Y., Bozdag, D., Ekici, E., Ozguner, F., and Lee, C. Partitioning based mobile element scheduling in wireless sensor networks. In Proceedings of the 2nd IEEE Conference on Sensor and Ad Hoc Communications and Networks, 2005.

- [YAO06] Gu, Y., Bozdag, D., and Ekici, E. Mobile element based differentiated message delivery in wireless sensor networks. In Proceedings of the 7th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, 2006.
- [ZAM03] W. Zhao and M.H. Ammar, Message ferrying: proactive routing in highly-partitioned wireless ad hoc networks, IEEE Workshop Future Trends of Distributed Computing Systems, 2003.
- [ZAZ05] W. Zhao, M. Ammar, and E. Zegura, Controlling the mobility of multiple data transport ferries in delay-tolerant network. IEEE INFOCOM, 2003.

Résumé Les data ferries sont utilisés principalement dans la collecte de données issues des capteurs par une communication à un seul saut. Cette stratégie de collecte est d'une importance capitale. Elle assure une optimisation énergétique considérable au sein d'un RCSF par la réduction des communications multi-sauts entre les nœuds capteurs et le sink. Le principal désavantage de cette approche de collecte est l'augmentation de la latence. L'élaboration d'un schéma optimal de collecte qui fait un compromis entre ces deux paramètres : l'énergie et la latence, est un défi intéressant. Tous les travaux de la planification de la trajectoire d'un data ferry modélisent le champ de communication par une forme circulaire appelée le modèle disque. Dans la réalité et de point de vue géométrique le champ de communication n'est pas réellement un disque, il n'y a pas une frontière nette entre les communications réussies et échouées. Ce mémoire, propose une nouvelle solution pour la planification de la trajectoire d'un data ferry basée sur un modèle de communication réaliste. Notre solution s'articule sur une nouvelle formulation du problème classique de planification basé TSPN par un problème d'optimisation multi-objectif pour l'optimisation des deux paramètres : énergie et latence. La résolution de ce problème est effectuée en plusieurs étapes. Dans la dernière étape, une fonction qui fait le compromis entre l'énergie et la latence pondérés par des paramètres est définie. Un ajustement de ces paramètres permet d'atteindre les objectifs souhaités. La solution modélise la région d'intérêt par une grille bidimensionnelle et des heuristiques basées TSP et SSP sont utilisées pour la planification de la trajectoire.

Mots clés : Réseau de capteurs, data ferries, collecte de données, modèle de communication probabiliste, planification de la trajectoire, SSP, TSP, TSPN.

Abstract Data ferries are used mainly in data gathering from sensors by a single-hop communication. This data gathering strategy is of capital importance. It ensures a considerable energy optimization within WSNs by the reduction of multi-hop communications between the sensor nodes and the sink. The principal disadvantage of this data gathering approach is the increase in latency. Elaborate an optimal data gathering scheme which makes a tradeoff between these two parameters: energy and latency, is an interesting challenge. All work of the data ferry path planning model the communication field by a circular form called the model disc. In reality and from geometrical point of view the communication field is not really a disc, there is not a clear border between the successes and failed communications. This work, proposes a new solution for the data ferry path planning based on a realistic communication model. Our solution is articulated on a new formulation of the classical planning problem based TSPN by a multi-objective optimization problem for the optimization of the two parameters: energy and latency. The problem resolution is carried out in several steps. In the last step, we define function which makes the tradeoff between the energy and latency balanced by parameters. An adjustment of these parameters makes it possible to reach the wished objectives. The solution models the area of interest by a two-dimensional grid and heuristics based TSP and SSP are used for trajectory planning.

Keywords: Wireless Sensor Network, data ferries, data gathering, probabilistic communication model, path planning, SSP, TSP, TSPN.