

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE  
UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE HAOUARI BOUMEDIENNE



FACULTE D'ELECTRONIQUE ET D'INFORMATIQUE

## Mémoire

Présenté pour l'obtention du diplôme de : **MAGISTER**  
Spécialité : **Informatique Mobile**

**Mr. ZAIR Mustapha**

## Thème

*Exclusion Mutuelle dans les Réseaux  
de Capteurs et Actionneurs Sans fil*

Soutenu publiquement, le : 14/07/2010 devant le jury composé de :

Mr. S. LARABI	Prof à l'USTHB	Président
Mr. A. DERHAB	Chercheur CERIST	Directeur de mémoire
Mr. M. BENCHAIBA	Docteur à l'USTHB	Examineur
Mme. S. MOUSSAOUI	Docteur à l'USTHB	Examineur

N° Réf : 15/2010-M/inf

*Promotion 2009-2010*

# Remerciements

*Je tiens à remercier notre bon dieu, le tout puissant de nous avoir permis de mener bien ce modeste travail.*

*Mes remerciements et nos profondes gratitude à mon promoteur **Mr A. DERHAB** de m'avoir proposé ce sujet, de m'avoir encadré, pour ces remarques pertinentes et ses conseils judicieux le long de ce travail.*

*Je tiens à remercier le personnel du Centre de Développement des Technologies Avancées.*

*Nos remerciements s'adressent également aux membres du jury d'avoir accepté d'évaluer notre travail.*

*Je suis reconnaissant envers tous ceux qui ont contribué de près ou de loin à la réalisation de ce projet.*

*En fin, mes remerciements envers tous mes amis qui m'ont soutenu durant la réalisation de ce projet.*

***Encore une fois merci à tous.***

# Dédicaces

*Rien n'est aussi beau à offrir que le fruit d'un labeur qu'on dédie du fond du cœur à ceux qu'on aime et qu'on remercie en exprimant la gratitude et la reconnaissance durant toute notre existence.*

**Je dédie ce modeste travail :**

- ❖ *Aux deux bougies de ma vie mes très chères parents en témoignage de leurs sacrifices, leurs dévouements et leurs soutiens à qui je dois tout !*
- ❖ *A mon très cher promoteur Monsieur A. DERHAB pour son aide, et sa compréhension.*
- ❖ *A mes très chers amis.*
- ❖ *A tous les membres de la famille ZAIR.*
- ❖ *A toute la famille des enseignants à l'USTHB.*
- ❖ *A tous mes collègues universitaires FAYCEL, BELKACEM, BASSEL, OUALID, ... et à tous les étudiants de la session 2008/2010 en souhaitant le bon courage à tous ceux qui travaillent pour bien de l'humanité.*

MUSTAQHA

# Tables des Matières

Introduction Générale .....	1
<b>CHAPITRE 1 : Généralités sur les WSANs</b>	
I. Introduction .....	3
II. Composants d'un WSANs .....	3
II.1. Micro-capteur .....	4
II.1.1. Caractéristiques physiques d'un micro-capteur .....	5
II.1.2. Architecture matérielle d'un micro-capteur .....	6
II.1.2.1. L'unité de capture .....	6
II.1.2.2. L'unité de traitement .....	7
II.1.2.3. L'unité de transmission .....	7
II.1.2.4. L'unité d'énergie .....	8
II.1.3. Type de capteur .....	8
II.2. Actionneur .....	9
II.2.1. Présentation .....	9
II.2.2. Architecture matérielle d'un actionneur .....	10
II.2.2.1. Unité de décision .....	11
II.2.2.2. Unité d'actuation .....	11
III. Architecture des WSANs .....	11
III.1. Architecture semi-automatisée .....	12
III.2. Architecture automatisée .....	13
IV. Applications .....	14
IV.1. Applications militaires .....	14
IV.2. Applications à la sécurité .....	15
IV.3. Applications environnementales .....	15
IV.3.1. Contrôle environnemental d'intérieur .....	15
IV.3.2. Applications à l'agriculture .....	16
IV.4. Applications commerciales .....	17
IV.5. Applications médicales et vétérinaire .....	17
IV.6. Applications de robotique .....	18
V. Pile protocolaire des WSANs .....	18
V.1. Le plan de gestion .....	19
V.1.1. Gestionnaire d'énergie .....	19
V.1.2. Gestionnaire de mobilité .....	19
V.1.3. Gestionnaire d'erreurs .....	20
V.2. Plan de coordination .....	20
V.3. Le plan de communication .....	21
V.3.1. La couche transport .....	21
V.3.2. La couche de routage .....	21
V.3.3. Medium Access control (MAC) .....	22
V.3.4. Cross-layering .....	23
VI. Conclusion .....	24

## **CHAPITRE 2 : L'exclusion mutuelle dans les réseaux WSANs**

I. Introduction .....	25
II. Le problème d'exclusion mutuelle .....	25
III. Classification des algorithmes d'exclusion mutuelle .....	26
III.1. L'exclusion mutuelle dans les réseaux statiques .....	26
III.2. L'exclusion mutuelle dans les réseaux cellulaires .....	26
III.3. L'exclusion mutuelle dans les réseaux mobiles Ad Hoc .....	27
III.4. L'exclusion mutuelle dans les WSANs .....	27
III.4.1. Exclusion mutuelle basée ressource .....	29
III.4.2. Exclusion mutuelle basée type de chevauchement .....	30
III.4.3. Exclusion mutuelle basée région de chevauchement .....	30
III.4.4. Exclusion mutuelle basée intensité de chevauchement .....	31
IV. Autres variantes d'événements .....	31
V. Conclusion .....	33

## **CHAPITRE 3 : Etat de l'art**

I. Introduction .....	34
II. La solution de Vedenham et al. ....	34
II.1. Approche centralisée .....	34
II.1.1. Description de l'approche .....	34
II.1.2. Exemple d'illustration .....	36
II.2. Approche distribuée .....	37
II.2.1. Définition .....	37
II.2.2. Description de l'approche .....	37
III. Etude de la solution de Vedenham et al .....	37
IV. Travaux liés au problème d'exclusion mutuelle dans les WSANs .....	39
IV.1. La solution de Tommaso Melodia et al. ....	39
IV.2. La solution de Akkaya et Younis .....	41
IV.2.1. Définition du problème .....	41
IV.2.2. Description de la solution .....	41
IV.3. La solution de McLaughlan et Akkaya .....	43
IV.3.1. Définition du problème .....	43
IV.3.2. Description de la solution .....	44
V. Conclusion .....	45

## **CHAPITRE 4 : CRMEA**

I. Introduction .....	47
II. Modèle du réseau .....	47
III. Formulation du problème .....	48
IV. Algorithme d'Exclusion Mutuelle Centralisé basé-Ressource (Centralised based-Ressource Mutual Exclusion Algorithme : CRMEA) .....	51
IV.1. La phase d'initialisation de CRMEA .....	51
IV.2. La phase d'optimisation du nombre d'actionneur dans la région d'évènement ..	52
IV.3. La phase d'optimisation du champ d'action dans la région d'évènement .....	53
IV.4. La phase d'optimisation par mobilité dans la région d'évènement .....	55
IV.5. Combinaison de l'optimisation du champ d'action et la mobilité .....	57
V. Conclusion .....	58

## *CHAPITRE 5 : Evaluation des performances*

I. Introduction .....	59
II. Métriques de performance .....	59
IV. Résultats .....	61
IV.1. Coût d'action .....	61
IV.2. Coût d'énergie .....	62
IV.3. Taux d'optimisation du coût .....	63
IV.4. Degré d'exclusion mutuelle .....	64
IV.5. Degré maximal de couverture .....	65
IV.6. Délai de mobilité .....	66
IV.7. Energie de mobilité .....	67
IV.8. L'effet de la vitesse sur le délai de mobilité .....	68
IV.9. L'effet de la vitesse sur l'énergie de mobilité .....	69
V. Conclusion .....	70
Conclusion générale .....	71
Bibliographie .....	72



## Table des figures

Figure 1 : L'architecture physique des WSANs .....	4
Figure 2: Micro-capteurs intégrés (Mica2dot) .....	5
Figure 3: Micro-capteurs carte (Fleck1) .....	5
Figure 4 : Architecture matérielle d'un micro-capteur .....	6
Figure 5: Tableau comparatif des standards IEEE .....	8
Figure 6 : Exemples de robots .....	10
Figure 7: Architecture matérielle d'un actionneur .....	11
Figure 8 : Les architectures d'un WSAN .....	12
Figure 9 : Modèle de protocole dans les WSANs .....	19
Figure 10 : Notations d'exclusion mutuelle .....	28
Figure 11 : Différentes régions basées sur la notation .....	29
Figure 12 : variantes d'évènements .....	32
Figure 13 : Exemple d'illustration de l'approche centralisée de Vedentham .....	36
Figure 14 : Exemple d'illustration de l'optimalité de la solution .....	38
Figure 15 : Partitionnement dans la région de l'évènement .....	40
Figure 16 : Régions chevauchées et non-chevauchées pour un collecteur c .....	40
Figure 17 : Redéploiement des actionneurs .....	42
Figure 18 : Exemples de : IS, DS, IDS, k-IDS .....	45
Figure 19 : La configuration de couverture des actionneurs .....	50
Figure 20 : L'organigramme général de CRMEA .....	51
Figure 21 : Exemple d'exécution de l'optimisation du champ d'action .....	55
Figure 22 : Exemple d'exécution de l'optimisation par mobilité .....	57
Figure 23 : Notations des stratégies de calcul de couverture .....	60
Figure 24 : Le coût d'action en fonction du rayon d'évènement .....	62
Figure 25 : Le coût d'énergie en fonction du rayon d'évènement .....	63
Figure 26 : Taux d'optimisation du coût en fonction du rayon d'évènement .....	64
Figure 27 : Degré d'exclusion mutuelle en fonction du rayon d'évènement .....	65
Figure 28 : Degré maximal de couverture en fonction du rayon d'évènement .....	66
Figure 29 : Délai de mobilité en fonction du rayon d'évènement .....	67
Figure 31 : Délai de mobilité en fonction de la vitesse des actionneurs (R=60m) .....	68
Figure 30 : Energie de mobilité en fonction du rayon d'évènement .....	68
Figure 32 : Energie de mobilité en fonction de la vitesse des actionneurs (R=60m) ..	69

## Liste des algorithmes

Algorithme 1 : Algorithme centralisé de Vedantham et al. ....	35
Algorithme 2 : La phase d'initialisation de CRMEA .....	52
Algorithme 3 : La phase d'optimisation du nombre d'actionneurs .....	53
Algorithme 4 : La phase d'optimisation du champ d'action dans R .....	54
Algorithme 5 : La phase d'optimisation par mobilité dans R .....	56

# Résumé

La tâche principale d'un réseau de capteurs sans fil (WSN) consiste à effectuer la capture sur l'environnement. Le besoin d'une interaction intelligente avec l'environnement a conduit vers l'émergence des réseaux de capteurs et actionneurs (WSANs) qui sont capables de faire des opérations de lecture et d'écriture (actionnement) sur l'environnement. L'évolution des WSNs vers les WSANs a introduit de nouveaux défis qui doivent être traités. Dans ce contexte, nous identifions le problème d'*exclusion mutuelle*, qui consiste à choisir l'ensemble minimal d'actionneurs pour couvrir une région d'évènement tout en minimisant le chevauchement entre les régions d'action des actionneurs sélectionnés. Nous classifions les différents types d'exclusion mutuelle dans les différents réseaux et plus précisément dans les WSANs. Nous montrons aussi les effets indésirables dans le cas où la couverture d'une région particulière n'est pas bonne. Nous présentons la seule solution dans la littérature qui traite ce problème, ensuite nous introduisons notre solution qui utilise une autre technique pour le choix des actionneurs de couverture et qui introduit des alternatives qui vont réduire encore le problème posé. A travers les simulations, nous étudions la performance de notre solution avec ces différentes extensions vis-à-vis la solution proposée dans la littérature, et nous montrons ainsi que notre solution est efficace et réduit considérablement le problème d'exclusion mutuelle dans les WSANs.

**Mots Clés :** réseau sans fil, capteurs, actionneurs, d'exclusion mutuelle, environnement, l'ensemble minimale, régions d'action, région d'évènement, chevauchement, couverture, effets indésirables.

# Abstract

A typical Wireless Sensor Network (WSN) performs only one action: sensing the environment. The need for smart interaction with the environment has led to the emergence of Wireless Sensor and Actor Networks (WSANs). The evolution from WSNs, which can be thought of to perform only read operations, to WSANs, which can perform both read and write operations, introduces unique and new challenges that need to be addressed. In this context, we identify the problem of *mutual exclusion*, which consist to find the minimal sub-set that covers the entire event region such that there is a minimal overlap in the acting regions. We define the different types of mutual exclusion in several cases of networks and especially in the context of WSANs, and show the undesirable consequences of not providing mutual exclusion with example applications. We introduce the unique solution in the literature which addresses this problem, than we propose our solution which select the cover sub-set differently, and introduce many alternatives that reduce considerably this problem. Through simulations, we study the performance of our solution with there different extensions in one hand, and the performance of the solution proposed in the literature in the other hand. We show that our solution is efficient for a variety of network conditions.

**Keywords:** Wireless Network, Sensor, sensing, Environment, Actor, mutual exclusion, minimal sub-set, cover, region, event, minimal overlap, acting regions.

# Introduction Générale

L'avancée technologique dans le domaine de la miniaturisation des composants électroniques et la communication sans fil a permis l'apparition d'un nouveau type de réseau appelé : *réseau de capteurs sans fil* ou *Wireless Sensor Networks* (WSNs) qui est composé d'un ensemble d'entités déployées de façon à couvrir un territoire donné. Ce dernier a une très grande variété d'applications dans le domaine civil, médical ou encore militaire. Cependant, les types d'actions réalisées par les WSNs sont limités uniquement à capter les grandeurs physiques d'un environnement (chaleur, humidité, vibration etc....).

L'intérêt grandissant pour une interaction intelligente avec l'environnement a engendré l'apparition d'une autre classe de réseaux, appelée : *Réseaux de capteurs et d'actionneurs sans fil* ou *Wireless Sensor and Actor Network* (WSAN), capables de capter des grandeurs physiques telles que les WSNs, mais aussi de réagir dans l'environnement en question par rapport aux données captées.

Dans les WSANs, en plus des nœuds capteurs, une classe spéciale de nœuds appelés actionneurs est ajoutée au réseau. Ces actionneurs sont capables d'agir sur l'environnement, ils sont riches en termes de capacités énergétiques, et peuvent être mobiles dans certains cas. Par exemple, en cas d'incendie, les capteurs peuvent envoyer l'origine et l'intensité de l'incendie aux actionneurs se trouvant à proximité afin que le feu soit rapidement éteint avant qu'il ne se propage. De même, des capteurs de lumière dans un bâtiment peuvent détecter la présence d'intrus, et guider des caméras pour le poursuivre. Les actionneurs sont équipés d'interfaces de communication sans fil, ce qui fait qu'ils sont capables soit de recevoir les commandes nécessaires à partir du nœud puits (architecture semi-automatisée), soit à partir des capteurs (architecture automatisée).

L'évolution des WSNs vers les WSANs introduit de nouveaux et uniques défis qui doivent être abordés, l'un de ces défis est comment choisir un ensemble d'actionneurs qui peut couvrir toute la région d'évènement, tout en minimisant les chevauchements entre les régions d'action. Dans la littérature, ce problème est appelée l'exclusion mutuelle dans les WSANs. Par exemple, considérons un

système d'arrosage automatique, avec des capteurs d'humidité et des arroseurs (actionneurs). Les arroseurs sont activés quand l'humidité dans une région se trouve au-dessous d'un certain seuil. Ici, il est préférable d'activer un ensemble d'actionneurs qui couvrent toute la région et minimisent les chevauchements entre les différentes régions d'action, c'est à dire minimiser les ressources d'actionneurs (eau) ainsi que l'énergie d'action nécessaires pour couvrir la région [1].

Ce mémoire est organisé comme suit :

Dans le premier chapitre, nous donnons des généralités sur les réseaux de capteurs et d'actionneurs (WSANs).

Dans le second chapitre, nous définissons le problème d'exclusion mutuelle dans les WSANs

Dans le troisième chapitre, nous présentons le seul algorithme d'exclusion mutuelle existant dans la littérature avec des travaux qui ont un apport avec notre problème.

Dans le quatrième chapitre, nous proposons un nouvel algorithme d'exclusion mutuelle basé ressource (*CRMEA*<sup>1</sup>) ainsi que des extensions qui optimisent plus les ressources d'actionneurs nécessaire pour couvrir la région d'évènement.

Dans le cinquième chapitre, nous comparons les performances de *CRMEA* et ses extensions avec celles de l'algorithme existant dans la littérature.

---

<sup>1</sup> *CRMEA* : Nous avons communiqué nos résultats de recherche dans le workshop suivant: Abdelouahid Derhab, and Mustapha Zair. "A Resource-based Mutual Exclusion Algorithm supporting Dynamic Acting Range and Mobility for Wireless Sensor and Actor Networks", In the First International Workshop on Mobility in Wireless Sensor Networks (MobiSensor 2010) in conjunction with the 6th IEEE/ACM International Conference on Distributed Computing in Sensor Systems (DCOSS '10), June 21-23, 2010

## I. Introduction

La diversité des problèmes liée à la perte de ressources critiques, les catastrophes naturelles et humaines, l'intrusion à des systèmes privés, et d'autres, a conduit à la recherche d'un nouveau type de réseaux à faible coût de déploiement qui est capable de détecter les événements et de réagir d'une manière adéquate. Les réseaux des capteurs et d'actionneurs sans fil (Wireless Sensor and Actor Network : WSAN) présentent une solution efficace. Le WSAN comme son nom l'indique est composé de plusieurs nœuds qui peuvent être soit des capteurs ou des actionneurs connectés avec des liaisons sans fil afin d'effectuer respectivement des tâches de capture et d'actionnement sur l'environnement pour annuler les événements indésirables. Dans, certains cas il est préférable (voir impératif) de déclencher les actions dans les plus courts délais. A cet effet, le processus de coordination entre les entités du réseau doit être performant afin de réagir à temps. Dans ce premier chapitre nous allons décrire ce type de réseau (fonctionnement, architecture, entités, et applications).

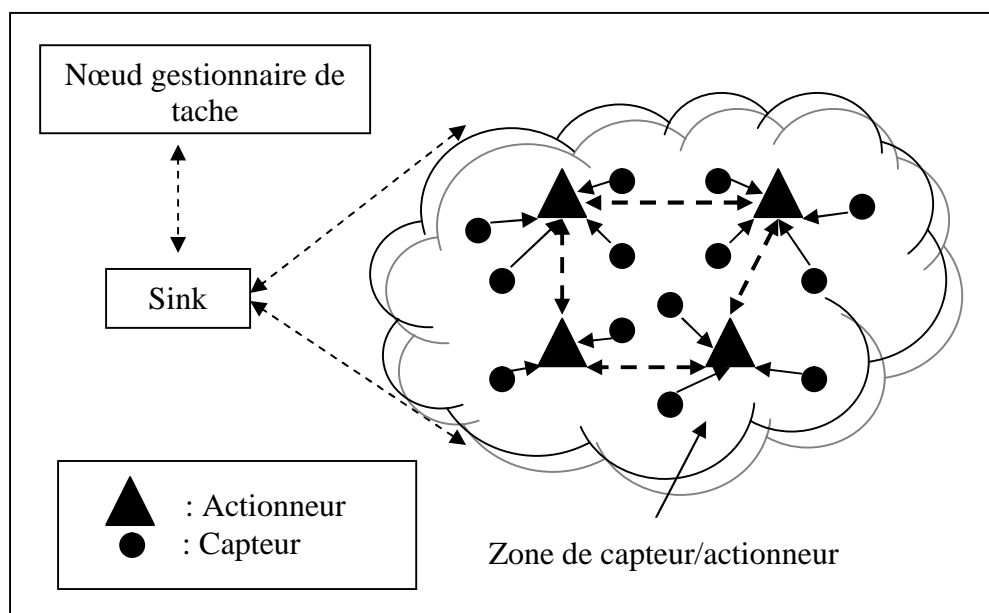
## II. Composants d'un WSAN

Un WSAN comme illustré à la figure 1, est un système distribué avec un grand nombre de nœuds qui sont soit des capteurs ou des actionneurs. Les capteurs sont des nœuds moins riches en termes d'énergie, portée de communication, capacité de calcul et de capture, leurs nombre est très important par rapport au nombre d'actionneurs qui sont généralement très coûteux par rapport aux capteurs.

La position des capteurs n'est pas obligatoirement prédéterminée. Ils peuvent être aléatoirement dispersés dans une zone géographique, appelée « *champ de captage* » correspondant à la région d'intérêt pour le phénomène capté. Cependant, la position des actionneurs est prédéterminée de façon à couvrir le maximum de la région d'intérêt, mais dans certaines applications les actionneurs sont mobiles.

Dans un WSAN, il existe un nœud spécial chargé de gérer les entités du réseau qui peuvent communiquer avec n'importe quel nœud dans le réseau, ce nœud est appelé nœud puits (ou *sink*). Ce dernier peut être connecté à l'utilisateur du réseau (via Internet, un [satellite](#) ou un autre système). L'utilisateur peut adresser des requêtes aux

autres nœuds du réseau, précisant le type de données requises et récolter les données environnementales captées par le biais du nœud puits.



**Figure 1 : L'architecture physique des WSANS**

## II.1. Micro-capteur

Les progrès réalisés ces dernières années dans les domaines de la [microélectronique](#) et des technologies de communication sans fil, ont permis de produire avec un coût raisonnable des composants de quelques millimètres cubes de volume, un peu plus grands qu'une pièce de monnaie (voir figure 2). Ces derniers, appelés micro-capteurs, intègrent : une unité de captage chargée de capter des grandeurs physiques (chaleur, humidité, vibrations) et de les transformer en grandeurs numériques ; une unité de traitement informatique et de stockage de données ; et un module de transmission sans fil.

Il existe deux approches pour construire un micro-capteur :

Micro-Capteurs intégrés : Ils intègrent l'unité de captage directement dans le micro-capteur, cette approche est plus économique et plus robuste, elle se traduit aussi par une diminution visible de la taille du micro-capteur (voir figure 2).

Micro-Capteurs de carte (Sensor Board) : L'unité de captage est connectable au Micro-Capteur à travers des bus d'extensions (voir figure 3), cette approche est plus à la demande de l'utilisateur qui peut connecter des capteurs de différents types suivant le domaine d'application sur le même micro-capteur [20].

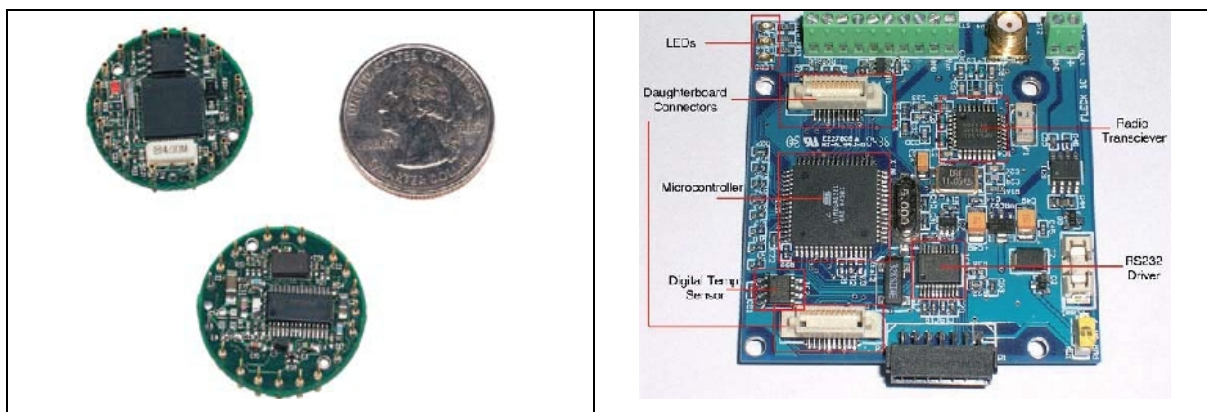


Figure 2: Micro-capteurs intégrés (Mica2dot) (Fleck1) Figure 3: Micro-capteurs carte

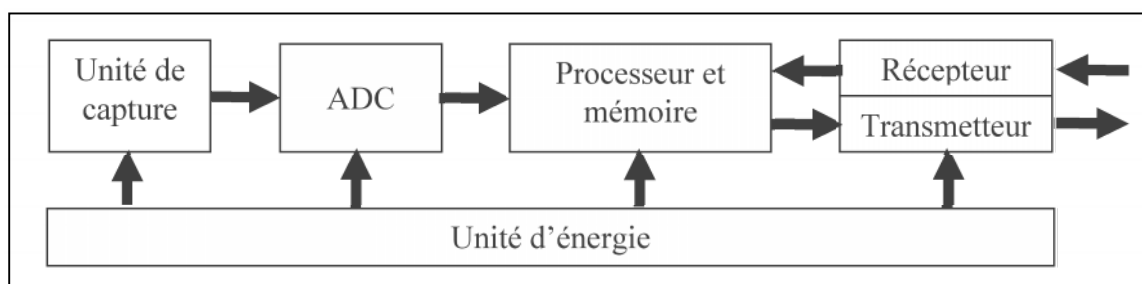
### II.1.1. Caractéristiques physiques d'un micro-capteur

- **Petite taille** : La taille d'un micro-capteur varie selon la composition entre  $1\text{cm}^3$  dans le cas d'un micro capteur intégré (voir figure 2), et  $4\text{cm} \times 4\text{cm}$  dans le cas d'un micro-capteur à carte avec des extensions possibles. Sa taille relativement petite lui permet de se loger dans beaucoup d'endroits avec discrétion.
- **Sources d'énergie très limitées** : L'énergie constitue le principal handicap des micro-capteurs, la taille de la batterie détermine la taille du capteur, même avec une batterie relativement grande (2 piles AAA) le besoin d'optimiser l'énergie surtout en communication est très importante.
- **Taux de transfert limité** : Un petit débit de données n'est pas handicapant pour un réseau de capteurs où les fréquences de transmission ne sont pas importantes. Le taux de transfert dans les micro-capteurs est limité à 250Kbps, taux théorique pour le protocole (ZigBee sur une fréquence de 2.4 GHz).
- **Faible coût de fabrication** : Le faible coût de production constitue un avantage considérable pour les micro-capteurs, ce qui pourra permettre de généraliser et diversifier leur utilisation, et pourquoi pas palier au défaut d'énergie par le faible coût de production.

- **Autonome et adaptatif** : Un micro-capteur intégré est complètement autonome, il peut effectuer toutes les tâches qui lui sont affectées (captage communication,...etc.) sans avoir besoin d'un composant tierce. En revanche, il est moins adaptatif qu'un micro-capteur à carte qui est très adaptatif et peut recevoir plusieurs capteurs pour différentes grandeurs physiques et ainsi il devient aussi autonome qu'un capteur intégré.

## II.1.2. Architecture matérielle d'un micro-capteur

Un nœud capteur contient quatre (04) unités de base : l'unité de captage, l'unité de traitement, l'unité de transmission, et l'unité de contrôle d'énergie [2]. Il peut contenir également, suivant son domaine d'application, des modules supplémentaires tels qu'un système de localisation ([GPS](#)), ou bien un système générateur d'énergie (cellule solaire). On peut même trouver des micro-capteurs, un peu plus volumineux, dotés d'un système mobilisateur chargé de déplacer le micro-capteur en cas de nécessité (voir figure 4).



**Figure 4 : Architecture matérielle d'un micro-capteur.**

### II.1.2.1. L'unité de capture

L'unité de captage est généralement composée de deux (02) sous-unités : le capteur lui-même et un convertisseur Analogique/Numérique (ADC). Le capteur est responsable de fournir 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.

### II.1.2.2. L'unité de traitement

L'unité de traitement comprend un processeur associé généralement à une petite unité de stockage et fonctionne à l'aide d'un système d'exploitation spécialement conçu pour les micro-capteurs ([TinyOS](#) par exemple) [21]. Cette unité est chargée d'exécuter les protocoles de communications qui permettent de faire collaborer le nœud avec les autres nœuds du réseau. Elle peut aussi analyser les données captées et les agréger pour un gain d'énergie dans la transmission.

### II.1.2.3. L'unité de transmission

Cette unité est responsable d'effectuer toutes les émissions et les réceptions des données sur un medium sans fil. Elle peut être de type optique ou le plus souvent de type radiofréquence.

Les communications de type optique sont robustes vis-à-vis des interférences électriques. Néanmoins, elles présentent l'inconvénient d'exiger une ligne de vue permanente entre les entités communicantes. Par conséquent, elles ne peuvent pas établir de liaisons à travers des obstacles.

Les unités de transmission de type radiofréquence comprennent des circuits de modulation, démodulation, filtrage et multiplexage ; ce qui implique une augmentation de la complexité et du coût de production du micro-capteur. La taille de la batterie détermine la taille du micro-capteur.

Un autre défi des micros-capteurs est comment faire la transmission à faible coût d'énergie. Deux (02) types de transmission sont utilisés à l'heure actuelle.

#### **a. CSMA de base :**

Utilisant les fréquences (433/868/916) Mhz pour une bande passante qui peut atteindre les 50Kpps et une distance de transmission jusqu'à 300m.

#### **b. ZigBee 802.15.4 :**

Destinée pour les périphériques à faibles ressources énergétiques, cette norme ratifiée en 2004 fonctionne sur les fréquences 868/914Mhz et 2,4Ghz avec un débit qui peut atteindre 250Kbps sur une distance de 100m, son principal

avantage est le coût énergétique très réduit et la rapidité de connexion des nœuds aux réseaux.

La figure 5 présente une comparaison des différents standards sans fil actuels :

Protocole	Zigbee	Bluetooth	Wi-Fi
IEEE	802.15.4	802.15.1	802.11a/b/g
Besoins mémoire	4-32 Kb	250 Kb +	1 Mb +
Autonomie avec pile	Années	Jours	Heures
Nombre de nœuds	65 000+	7	32
Vitesse de transfert	250 Kb/s	1 Mb/s	11-54-108 Mb/s
Portée	100 m	10-100 m	300 m

**Figure 5: Tableau comparatif des standards IEEE**

#### II.1.2.4. L'unité d'énergie

Un micro-capteur est muni d'une ressource énergétique (généralement une batterie) pour alimenter tous ses composants. Cependant, en conséquence de sa taille réduite, la ressource énergétique dont il dispose est limitée et généralement irremplaçable. Dès lors, l'énergie est la ressource la plus précieuse dans un réseau de capteurs et d'actionneurs, puisqu'elle influe directement sur la durée de vie des micro-capteurs et du réseau en entier.

L'unité de contrôle d'énergie constitue donc l'un des systèmes les plus importants. Elle est responsable de répartir l'énergie disponible aux autres modules et de réduire les dépenses en mettant en veille les composants inactifs par exemple. Cette unité peut aussi gérer des systèmes de rechargement d'énergie à partir de l'environnement observé (power generator) telles que les cellules solaires, afin d'étendre la durée de vie totale du réseau.

#### II.1.3. Type de capteur

Différents types de capteurs peuvent se greffer sur un micro-capteur pour étendre ses capacités tels que :

Capteurs de distance : ultrason, micro-onde, triangulation...

Capteurs de lumière : photodiode, capteur photographique...

Capteurs de sons : microphone, hydrophone...

D'autres capteurs de température, pression, débit ou encore d'inertie (pour le calcul de l'accélération) sont tous des capteurs qui peuvent collaborer en réseau dans divers domaines d'applications.

## II.2. Actionneur

### II.2.1. Présentation

Dans les WSANS, les phénomènes de capture des grandeurs physiques et de réaction dans l'environnement sont réalisés par les micro-capteurs et les actionneurs, respectivement. De ce fait, les actionneurs sont des nœuds richement équipés avec une plus grande capacité de calcul et de communication à longue distance, tout en gardant une durée de vie de batterie plus longue. Ainsi un actionneur est beaucoup plus cher qu'un micro-capteur d'où le déploiement des micro-capteurs est 10 à 100 fois plus important que le nombre d'actionneurs.

Deux (02) notions sont à éclaircir :

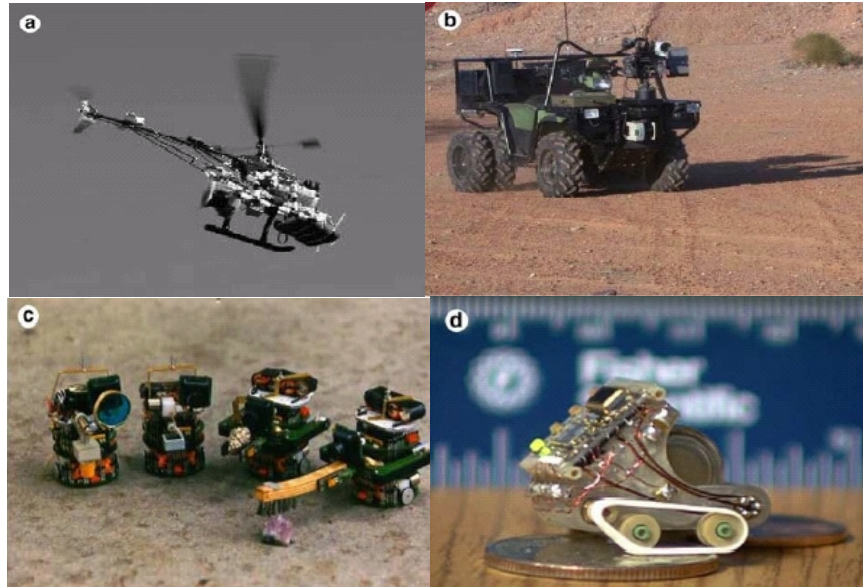
Actuator: Périphérique qui peut convertir un signal de contrôle électrique en une action physique. Il constitue le mécanisme avec lequel l'agent réagit sur l'environnement physique.

Actor ou Actionneur: En plus d'être capable de réagir dans un environnement en utilisant un ou plusieurs Actuator, il est aussi une entité réseau qui peut effectuer des opérations de communication, c'est-à-dire, recevoir et transmettre des données [22].

Dans quelques applications, des nœuds capteurs/actionneurs intégrés peuvent remplacer les nœuds actionneurs car ils peuvent effectuer les tâches des capteurs et des actionneurs à la fois, notons que ces nœuds comportent un ADC (convertisseur analogique vers numérique) en plus des composants d'un actionneur.

Un des exemples des nœuds capteurs/actionneurs est le robot [2], les robots ont fait l'objet de recherche de plusieurs laboratoires comme indiqué à la figure 6(a)-(d). La plateforme de l'hélicoptère montrée à la figure 6(a) fournit des cartes de terrain,

et une coopération air-à-terre avec des robots véhiculaires autonomes. Pour cela, il est vraisemblable que cet hélicoptère supporte des fonctionnalités comme l'arrosage avec l'eau ou la disposition de gaz.



**Figure 6 : Exemples de robots : (a) hélicoptère de trace aérien, (b) mule robotique, (c) télé-robots intelligents de sous-kilogramme (d) mini-robot**

Un exemple de robot de mule est montré dans la figure 6(b), il est conçu pour fonctionner dans les champs de bataille. Ces robots peuvent détecter et marquer des mines, transporter les armes, fonctionne comme des écharts ou peuvent dans l'avenir remplacer les soldats carrément.

En plus, SKITS montré à la figure 6(c) sont des robots en réseau à distance ayant une tourelle de radio permettant une communication sur fréquence UHF à 4800 Kbits/sec. Ces robots peuvent coordonner avec chaque autre en exploitant leurs capacités de communication sans fil et effectuer des tâches déterminées par l'application.

Dans la figure 6(d), on voit une image sur le plus petit robot (1/4 cube d'inch et pesant moins d'une once) développé dans les laboratoires national de Sandia.

## II.2.2. Architecture matérielle d'un actionneur

Un actionneur comme illustré à la figure 7 contient en plus des unités de transmission, de contrôle d'énergie et de traitement qui sont bien plus puissantes que

sur les micro-capteurs, une unité de décision ou le contrôleur et une unité de réaction (actuation unit).

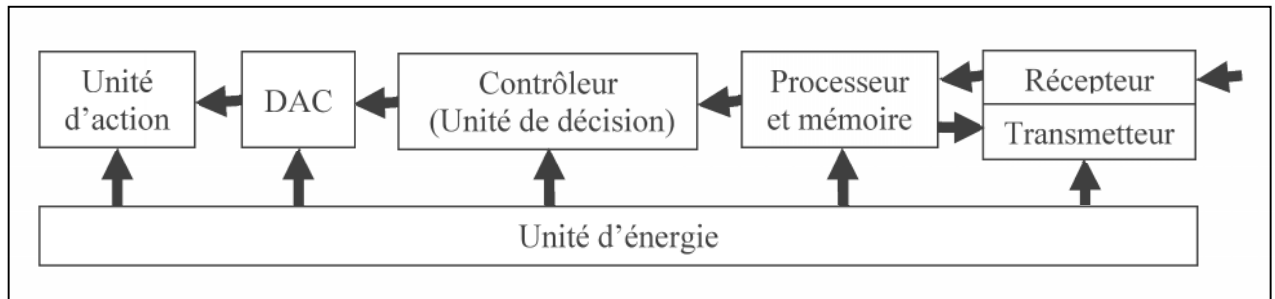


Figure 7: Architecture matérielle d'un actionneur

### II.2.2.1. Unité de décision

C'est l'unité chargée de lancer les actions sur l'unité d'actuation, elle permet de prendre des décisions d'action, du type d'action et sur quelle unité d'actuation en traitant les données reçues de plusieurs micro-capteurs.

### II.2.2.2. Unité d'actuation

Un actionneur peut disposer d'une ou de plusieurs unités d'actuation (*actuator*) suivant le domaine d'application, chaque unité d'actuation contient un convertisseur Numérique/Analogique (DAC) qui convertit le signal et le transmet à l'*actuator* en charge d'exécuter l'action.

## III. Architecture des WSANs

Après la détection d'un évènement dans l'environnement, les capteurs envoient d'une manière distribuée, autonome, et grâce à un routage multi-saut l'information aux actionneurs ou au sink, qui rassemblent, traitent, et reconstruisent éventuellement ces informations. Dès la détection de l'évènement, les actionneurs communiquent entre eux afin de prendre des décisions appropriées pour déclencher les actions nécessaires.

Les WSANs ont les caractéristiques suivantes qui sont propres à ce type de réseau :

- Temps réel : Dans les WSANS, selon la nature de l'application, il peut être nécessaire de déclencher l'action le plus vite possible. Par exemple, dans une application de lutte contre le feu, en détectant un évènement le déclenchement de l'action doit se faire le plus rapidement avant que le feu se propage. De plus, les données collectées et délivrées par les capteurs doivent être encore valides pendant l'exécution de l'action.
- Coordination : Contrairement aux WSNs où l'unité centrale (sink) effectue la collection des données, dans les WSANS, des nouveaux phénomènes qui surgissent : nous parlons de la coordination capteur-actionneur et la coordination actionneur-actionneur.

Dans les WSANS, il existe deux types d'architectures : semi-automatisée et automatisée. Ces deux architectures sont représentées dans la figure 8.

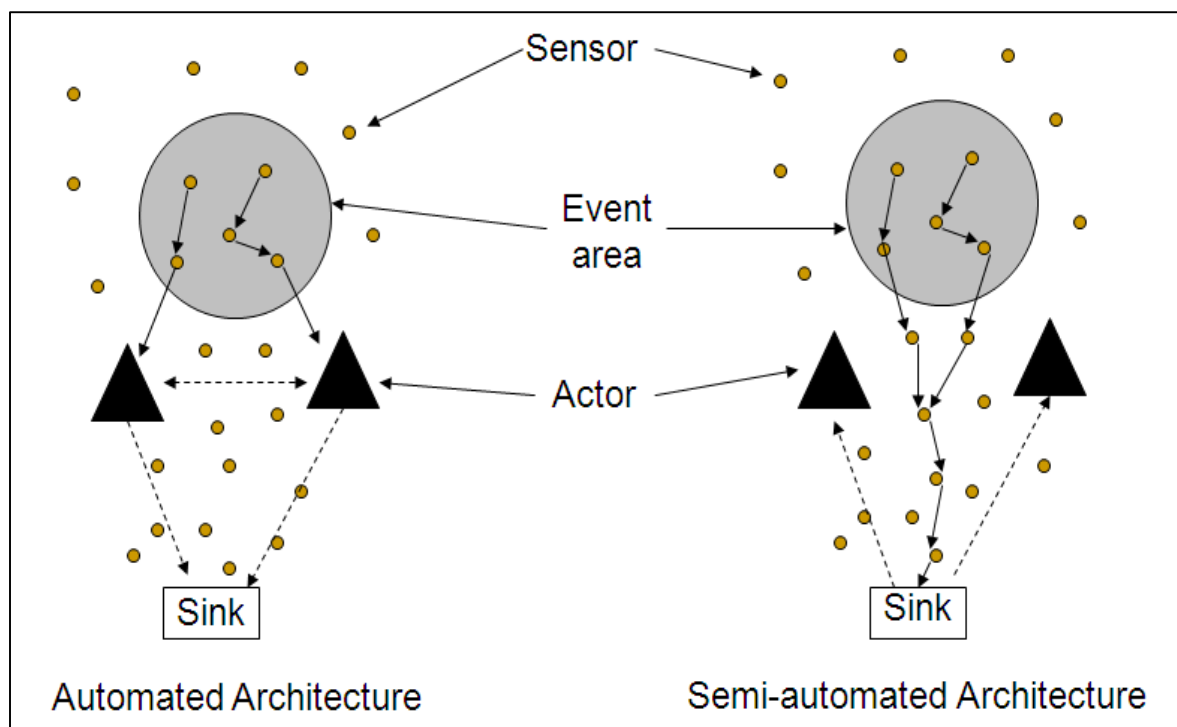


Figure 8 : Les architectures d'un WSAN

### III.1. Architecture semi-automatisée

L'architecture semi-automatique est basée sur un contrôleur central appelé la source (Sink) en d'autres termes tout doit passer par la source, collecter les données des différents micro-capteurs et communiquer les commandes à exécuter aux

actionneurs, ce rôle de coordinateur nécessite beaucoup de ressources (puissance de calcul, énergie...), c'est pour cette raison que la source est généralement fixe et richement équipée.

Cette architecture est similaire à celle adoptée pour les réseaux de capteurs (WSN), elle ne prend pas en compte les actionneurs [2]. Ainsi, il n'y a pas besoin de développer de nouveaux algorithmes et protocoles de communications et de coordinations.

Du fait qu'elle ne prend pas en compte l'hétérogénéité du réseau, cette architecture peut engendrer une charge de communication importante sur les micro-capteurs et peut causer ainsi la déconnexion du réseau.

### III.2. Architecture automatisée

Dans ce type d'architecture, les micro-capteurs se chargent de transmettre les données aux actionneurs, ces derniers disposent d'une durée de vie et d'une distance de transmission plus grande. Ils peuvent suivant le domaine d'application transmettre efficacement ces données à la source ou encore se coordonner entre eux pour exécuter une action bien définie.

Cette approche est plus efficace essentiellement sur deux (02) points importants :

Délai de latence court : Les données captées sont acheminées du micro-capteur à l'actionneur instantanément (voir la figure 8). Ainsi la latence est minimisée dans le cas d'une architecture automatique.

Durée de vie du réseau plus grande : L'architecture semi-automatique peut causer la déconnexion du réseau. Due au fait que la charge de communication des nœuds à un saut de la source sont souvent trop sollicités. Dans une architecture automatique différents évènements sont rapportés à différents actionneurs. En plus que les actionneurs peuvent être mobiles, ce qui diminue la charge de communication pour les micro-capteurs à un saut d'un actionneur et assure une durée de vie plus longue au réseau.

## IV. Applications

Les réseaux de capteurs et d'actionneurs sont constitués de plusieurs types de capteurs : sismiques, magnétiques, thermiques, visuelle, infrarouges, acoustiques et radars, qui sont capable de gérer une très grande variété de conditions ambiantes qui inclus les suivants :

Température, humidité, mouvement de véhicule, condition d'éclairage, la pression, le bruit, la présence ou l'absence de quelques objets, la pression mécanique sur les objets attachés et les caractéristiques courantes tel que : la vitesse, la direction, et la taille d'un objet.

Les nœuds capteurs peuvent être utilisés pour la capture continue, la détection d'évènement, identifier l'évènement, et le contrôle locale des actionneurs. Le concept de micro-capture et la connexion sans fil de ces nœuds promettent de nouveaux domaines d'applications. Nous catégorisons les applications en : militaire, environnement, santé, maison et autres applications commerciales. Il est possible d'enrichir cette classification avec d'autres catégories comme : l'exploration d'espace, traitement chimique, les opérations de secours en cas de catastrophes [4].

### IV.1. Applications militaires

Comme dans le cas de plusieurs technologies, le domaine militaire a été un moteur initial pour le développement des réseaux de capteurs et d'actionneurs. Le déploiement rapide, le coût réduit, l'auto-organisation et la tolérance aux pannes des WSANS sont des caractéristiques qui rendent ce type de réseaux un outil appréciable dans un tel domaine. Certaines de ces applications peuvent s'automatiser avec l'utilisation d'actionneurs. Par exemple, la détection des mines qui peut être effectuée par des actionneurs, et ainsi éviter les pertes humaines causées par ces mines.

Ces mêmes mines anti-personnelles peuvent être jugées trop dangereuses et moins efficaces dans un avenir proche, et pourraient être remplacées par un réseau de capteurs et d'actionneurs capables non seulement de détecter l'ennemi et de le capturer, mais aussi, de le tracer et de le poursuivre.

## IV.2. Applications à la sécurité

Les structures d'avions, navires, automobiles, métros, etc pourraient être suivies en temps réel par les WSANs, de même que les réseaux de circulation ou de distribution de l'énergie. Les altérations de structure d'un bâtiment, d'une route, d'un quai, d'une voie ferrée, d'un pont ou d'un barrage hydroélectrique (suite à un [séisme](#) ou au vieillissement) pourraient être détectées par des capteurs préalablement intégrés dans les murs ou dans le béton, sans alimentation électrique ni connexions filaires. Certains capteurs ne s'activant que périodiquement peuvent fonctionner durant des années, voire des décennies. Un WSAN de mouvements peut constituer un système d'alarme distribué qui servira à détecter les intrusions sur un large secteur. Déconnecter le système ne serait plus aussi simple, puisqu'il n'existe pas de point critique. La surveillance de routes ou voies ferrées pour prévenir des accidents avec des animaux ([roadkill](#)) ou des êtres humains ou entre plusieurs véhicules est une des applications envisagées des WSANs.

Selon leurs promoteurs, ces réseaux pourraient diminuer certaines failles de systèmes de sécurité et mécanismes de sécurisation, tout en diminuant leur coût. D'autres craignent aussi des dérives sécuritaires ou totalitaires si l'usage de ces réseaux n'est pas assujéti à des garanties éthiques sérieuses.

## IV.3. Applications environnementales

Le contrôle environnemental représente une catégorie très importante. On peut retenir deux sous-classes d'applications :

### IV.3.1. Contrôle environnemental d'intérieur

Un bon nombre de micro-capteurs peut trouver sa place dans le contrôle de l'environnement d'intérieur tel que les capteurs : de températures, de lumières, de pollution et d'humidité. Ces derniers peuvent servir à informer des actionneurs comme l'extincteur d'incendie ou le contrôleur de la climatisation. L'application des

WSANs peut être très bénéfique surtout si on sait qu'une grande partie de l'énergie est gaspillée, faute d'éclairer ou de climatiser un endroit vide ou avec une fenêtre ouverte.

Des études ont montré que le contrôle de l'éclairage et de la climatisation dans les grands buildings peut diminuer la consommation d'énergie par un facteur de deux *quatrillion*<sup>1</sup> de BTU<sup>2</sup> et économisera ainsi 55 milliards de dollars par an, mais aussi diminuer l'émission de gaz carbonique dans l'air par 35 million de tonnes.

Les systèmes de détection d'incendie sont courants et même obligatoires dans les grands buildings. L'application des WSANs peut non seulement coopérer pour éteindre un incendie plus rapidement et plus efficacement, mais aussi coordonner l'évacuation des occupants pris au piège en trouvant le meilleur chemin non obstrué par l'incendie ou le tremblement de terre et ainsi sauver leurs vies.

D'autre part, les applications de sécurité des WSANs qui peuvent détecter une fuite de gaz ou une intrusion, l'actionneur peut couper l'arrivée du gaz et ventiler la pièce, ou dans l'autre cas fermer les sortie et éjecter un gaz tranquilisant.

### IV.3.2. Applications à l'agriculture

Le déploiement d'un WSANs dans l'agriculture peut permettre d'augmenter la production et améliorer la qualité des produits. La capture des informations sur l'humidité, la pluie, la température de l'air et du sol, qui peuvent être transmises à des actionneurs qui réagissent pour assurer des conditions idéales pour le type de produit en exploitation.

Les WSANs peuvent aussi se révéler très économiques en gérant les systèmes d'irrigation, surtout que dans le domaine de l'agriculture, l'eau est une ressource très importante.

---

<sup>1</sup> *Quatrillion* : Un million de trillions, soit  $10^{24}$

<sup>2</sup> 1 BTU= 1055 joules

#### IV.4. Applications commerciales

Des nœuds capteurs pourraient améliorer le processus de stockage et de livraison (pour garantir la chaîne du froid en particulier). Le réseau ainsi formé, pourra être utilisé pour connaître la position, l'état et la direction d'un paquet ou d'une cargaison. Un client attendant un paquet peut alors avoir un avis de livraison en temps réel et connaître la position du paquet. Des entreprises manufacturières, via des WSANs pourraient suivre le procédé de production à partir des matières premières jusqu'au produit final livré. Grâce aux WSANs, les entreprises pourraient offrir une meilleure qualité de service tout en réduisant leurs coûts. Les produits en fin de vie pourraient être mieux démontés et recyclés ou réutilisés si les microcapteurs en garantissent le bon état. Dans les immeubles, le système domotique de chauffage et climatisation, d'éclairage ou de distribution d'eau pourrait optimiser son efficacité grâce à des micro-capteurs présents dans des tuiles au plancher en passant par les murs, huisseries et meubles. Les systèmes ne fonctionneraient que là où il faut, quand il faut et à la juste mesure. Utilisée à grande échelle, une telle application permettrait de réduire la demande mondiale en énergie et indirectement les émissions de [gaz à effet de serre](#). Rien qu'aux États-Unis, cette économie est estimée à 55 milliards de dollars par an, avec une diminution de 35 millions de tonnes des émissions de carbone dans l'air. Le monde économique pourrait ainsi diminuer ses impacts environnementaux sur le climat.

#### IV.5. Applications médicales et vétérinaire

La médecine peut aussi profiter de l'application des WSANs. Les capteurs peuvent capter des paramètres tel le rythme cardiaque et le taux de sucre dans le sang et transmettre ces informations à des actionneurs qui interviendront avant que la situation ne devienne critique.

La surveillance des enfants est aussi une application possible où les capteurs peuvent s'embarquer dans des jouets, ce qui permettra de surveiller le comportement des enfants, prévenir et réagir en conséquence.

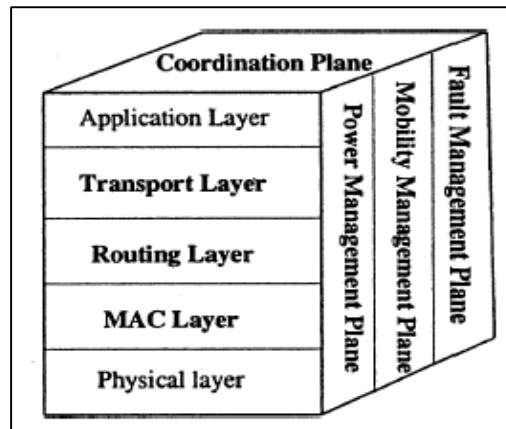
## IV.6. Applications de robotique

Beaucoup d'applications de robotique utilisent les WSANs, c'est un domaine d'application naturel où les robots eux-mêmes peuvent être des actionneurs. La robotique représente une réelle automatisation des WSANs. Nous pouvons imaginer toutes sortes de robots capables de réagir dans l'environnement en se basant sur des informations captées par des capteurs distribués ou même embarqués sur le robot lui-même.

L'utilisation des WSANs poursuit un vieux rêve de l'humanité, un environnement intelligent qui réagit automatiquement aux phénomènes physiques observés. Cette technologie répond à un enjeu économique très important, d'abord par son coût de déploiement très faible et son gain très important en énergie et autres ressources critiques.

## V. Pile protocolaire des WSANs

A ce jour, il n'existe pas une pile protocolaire standardisée pour les WSNs et les WSANs. Ian F. Akyildi et al. [2] suggèrent que la pile protocolaire pour les nœuds capteurs et actionneurs consiste essentiellement de trois plans, (c-à-d , plan de communication, plan coordination, et plan gestion) montré à la figure 9, le plan de communication qui est constitué de cinq sous plans (couches) nommés : couche application, transport, routage, MAC, et physique permet l'échange d'information entre les nœuds du réseau. Les données reçues par un nœud au plan de communication sont délivrées au plan coordination qui décide comment le nœud acte sur les données reçues. En plus, le plan de coordination assure que les nœuds sont modélisés comme une entité sociale, c'est-à-dire en termes de coordination et négociation de techniques qu'ils possèdent. Le plan de gestion est responsable pour la planification et le contrôle des nœuds pour qu'ils opèrent proprement. Il fournit aussi l'information nécessaire pour le plan coordination pour effectuer les décisions.



**Figure 9 : Modèle de protocole dans les WSANS**

Dans les trois sous-sections suivantes, nous discutons les besoins et les caractéristiques de chaque plan pour les coordinations capteur-actionneur et actionneur-actionneur.

## **V.1. Le plan de gestion**

Il est constitué de trois sous plans suivants:

### **V.1.1. Gestionnaire d'énergie**

Il doit définir comment un nœud utilise son énergie. La gestion de l'énergie diffère d'un nœud capteur à un nœud actionneur, mais aussi d'un nœud routeur à un nœud décideur et d'un nœud capteur fixe à un nœud actionneur mobile. Par exemple, si le niveau d'énergie est bas, le gestionnaire d'énergie peut informer le plan de coordination que ce nœud ne peut pas agir sur l'environnement, router les données ou capter les évènements.

### **V.1.2. Gestionnaire de mobilité**

Il doit détecter et enregistrer les mouvements du nœud de manière à toujours maintenir la connectivité du réseau, ce qui n'est pas facile surtout pour les nœuds capteurs par défaut d'énergie et de passage d'échelle.

### V.1.3. Gestionnaire d'erreurs

Il doit garantir l'intégrité du nœud, il détecte et résout les problèmes. Si un nœud capte des données erronées, le gestionnaire d'erreurs doit les détecter et informer le plan de coordination. Si le nœud est victime d'un dysfonctionnement dans la transmission ou la réception des données, c'est le gestionnaire d'erreurs qui est chargé de le détecter et doit en informer le plan de coordination.

### V.2. Plan de coordination

Détermine comment un nœud agit selon les données reçues à partir du plan de communication et du plan de gestion.

Après la capture d'un événement, les capteurs communiquent leurs lectures entre eux. Au niveau de chaque nœud capteur, ces données échangées sont transférées au plan de coordination pour prendre décisions. De cette façon, les capteurs sont capables de coordonner entre eux dans une tâche de capture à haut niveau. En plus, la coordination capteur-capteur peut être exigée pour déterminer les nœuds qui ne transmettent pas les données (dû à l'énergie faible ou le protocole MAC appliqué), pour effectuer le routage multi-saut et l'agrégation des données et le plus important pour sélectionner l'actionneur(s) à qui les données capturées vont être transmises.

L'existence du plan de coordination peut être plus critique pour les actionneurs que pour les capteurs, car les actionneurs ont besoin de collaborer entre eux dans le but d'effectuer des actions appropriées.

Quand un événement se produit, le but commun de tous les actionneurs est de fournir les actions demandées pour cet événement, c'est-à-dire des capacités sociales (des capacités de négociation et de coordination sophistiquées) sont nécessaires dans les WSANS pour assurer un comportement cohérent dans l'ensemble des actionneurs. Ces capacités sociales requises d'un actionneur sont définies dans le plan de coordination. Spécifiquement, quelle couche dans la coordination actionneur-actionneur est responsable de prendre les décisions, sur quels actionneurs vont réagir, et sur quelle partie de la région de l'événement et est ce que ces actionneurs agissent en même temps ou séquentiellement et dans quel ordre ?

### V.3. Le plan de communication

Le plan de communication reçoit les commandes du plan de coordination selon cette information il établit les liens entre les nœuds en utilisant les protocoles de communication. Spécifiquement, le plan de communication traite : la construction des canaux physiques, l'accès du nœud au support de transmission (MAC), la sélection des chemins de routage et le transport des paquets d'un nœud à un autre.

Dans ce qui va suivre, on va voir les exigences et les défis des couches : transport, MAC et ceux du routage, et aussi l'intégration cross layer entre ces couches pour les communications capteur-actionneur et actionneur-actionneur.

#### V.3.1. La couche transport

En plus de la fiabilité conventionnelle, les nouveaux protocoles de transport doivent aussi supporter les exigences temps-réel dans WSANs. Plusieurs protocoles de couche de transport ont été développés pour les réseaux ad-hoc et WSNs dans les dernières années.

Cependant selon notre connaissance, il n'existe aucun protocole de transport efficace qui prend en charge la fiabilité et le temps-réel dans les WSANs à ce moment. Par exemple, quand le protocole de transport pour la communication capteur-actionneur détecte une faible fiabilité, le protocole de transport pour la communication actionneur-actionneur règle le trafic entre les actionneurs tel que l'actionneur recevant cette information détectée peut informer les actionneurs voisins de cette situation le plus tôt possible. Puisque les communications capteur-actionneur et actionneur-actionneur se produit consécutivement dans WSANs un protocole de transport unifié est nécessaire qui fonctionne bien pour les deux cas.

#### V.3.2. La couche de routage

Dans les WSANs, quand les capteurs détectent un événement, il n'y a aucun actionneur spécifique à qui le message va être envoyé. Cette incertitude est due à l'existence de plusieurs actionneurs ce qui cause des défis dans les solutions de routage.

Premièrement, sélectionner un actionneur est un défi pour un capteur source. Ensuite, les données sources doivent être routées vers l'actionneur sélectionné dans une énergie efficace. Tant que les données sources sont transmises à travers les capteurs de relais vers un noeud actionneur, elles peuvent être agrégées ou acheminées pour atteindre une grande efficacité.

En plus de déterminer la route sélectionnée et les données délivrées, le protocole de routage doit supporter les communications en temps réel en considérant les différentes dates limites dues aux différents intervalles de validité.

En outre, le protocole de routage doit aussi considérer le problème de priorité et doit fournir les données avec les moindres délais bornés pour aboutir à l'actionneur à temps.

Dans les dernières années, un effort considérable de recherches sur les problèmes de routage dans les WSNs[1].

Parmi les protocoles réalisés concernant le routage dans les WSANs, nous citons : DSR, AODV, OLSR [31], qui sont adaptés pour la communication actionneur-actionneur que pour les réseaux adhoc, pour la communication capteurs-actionneur nous citons SEAD [32], et SPEED [33].

### V.3.3. Medium Access control (MAC)

Pour transmettre effectivement (sans collisions) l'information d'un événement à partir de plusieurs capteurs aux actionneurs, il y a un besoin d'un protocole MAC. En plus, dans quelques applications (de robots distribués par exemple), les actionneurs peuvent être mobiles. Quand ils se déplacent, ils quittent les régions de transmission de quelques capteurs et entrent dans d'autres régions de capteurs.

Parmi les solutions proposées, nous citons : TRACE [23], PBP, ainsi qu'un protocole développé dans [24] qui est sans collision, il réduit le délai et fournit des garanties de temps réel, et garde l'énergie en éliminant les collisions.

Pour la communication actionneur-actionneur, les protocoles MAC existants et développés pour les réseaux adhoc ne peuvent pas être utilisés directement. Ils doivent supporter le trafic en temps réel.

### V.3.4. Cross-layering

Les protocoles actuels des WSANs et WSNs sont basés largement sur une approche en couches. Or, la sous-efficacité et l'inflexibilité de ce paradigme cause des pauvres performances pour WSANs, due aux contraintes de faible consommation d'énergie et la faible latence. Pour cela, au lieu d'avoir des couches individuelles, on a besoin d'un cross-layering où les couches sont intégrées entre elles (elles communiquent entre elles pour effectuer un intérêt commun).

Dans WSANs, parmi les facteurs principaux qui causent une faible fiabilité est la congestion des réseaux, la couche MAC réagit localement en faisant un retour exponentiel [25], tandis que la couche transport réagit en retardant les transmissions des capteurs. Cependant ces deux couches travaillent indépendamment l'une de l'autre, ce qui cause de l'inefficacité due à la duplication de fonctions. Par l'approche cross-layering chaque protocole partage ses données avec les autres protocoles ce qui évite ces inefficacités. Par exemple : dans les WSANs quand la congestion est grande, premièrement la couche MAC réagit sur cette congestion. Si cette réponse n'est pas suffisante, la couche MAC informe la couche de routage sur cette congestion, cette dernière informe le plan de coordination sur cette situation. Alors le plan de coordination et la couche de routage permettent le reroutage vers un autre noeud actionneur approprié, et si cet actionneur et les routes n'existent pas, le protocole de transport peut être utilisé pour figer le trafic des transmissions.

Un autre exemple de cross-layering dans WSANs est celui de l'optimisation de la taille des paquets transmis du capteur à l'actionneur, dans ce cas les couches de routage, MAC, et physique doivent fonctionner ensemble.

En plus des interactions entre les couches transport, MAC, physique et de routage, dans les WSANs, il est nécessaire d'avoir une interdépendance entre la couche application et ces couches. Bien que le réseau fournisse la meilleure QoS aux applications, cette QoS varie avec le temps autant que les conditions des canaux et la topologie du réseau changent. Pour cela, la couche application doit s'adapter à la QoS offerte.

Les idées citées précédemment sont encore valides pour la communication actionneur-actionneur. Cependant, les actionneurs peuvent être mobiles, les

caractéristiques des liaisons et la topologie du réseau changent rapidement. En cas d'une liaison perturbée qui est causée par quelque chose difficile à corriger dans la couche physique, c'est-à-dire une grande mobilité des nœuds, il est mieux pour la couche physique d'interagir avec les couches supérieures [26]. Par exemple, dans WSANs, les actionneurs effectuent des communications unicast au lieu du broadcast pour éviter que les capteurs reçoivent des messages inutiles. Donc, dans l'approche cross-layering, chaque couche de la pile protocolaire ne répond pas seulement aux variations locales, mais aussi répond aux informations venantes des autres couches [26,27].

## VI. Conclusion

Dans ce chapitre, nous avons donné un petit aperçu sur les réseaux sans fil de capteurs et d'actionneurs. Ces derniers sont très utiles dans différents domaines d'application (militaire, contrôle environnemental, support logistique, santé, ...etc.) depuis qu'ils sont à coût raisonnable, facile à déployés. Dans les WSANs, deux architectures de communication sont possibles : semi-automatisée et automatisée. Chaque architecture présente des avantages et des inconvénients. La rapidité, l'exactitude, la minimisation des ressources, et d'autres facteurs peuvent influencer sur le choix d'une architecture sur l'autre.

La migration de ce nouveau type de réseau dans la vie courante a introduit de nouveaux défis qui doivent être traités, parmi ces défis, on s'intéresse dans notre mémoire au problème de minimisation de ressources consommées par les actionneurs; ce problème sera détaillé dans le chapitre suivant.

## I. Introduction

L'évolution des WSNs en WSANs, a conduit à la présence de nouveaux défis liés à l'intégration des actionneurs. Parmi ces défis, nous traitons dans notre mémoire le défi de « comment optimiser l'utilisation des ressources d'actionneurs pour réagir sur un évènement donné en évitant des problèmes de chevauchement entre les actions déclenchées par les actionneurs ».

Dans les réseaux WSANs, si plusieurs actionneurs sont sélectionnés afin d'agir sur un évènement donné, il est nécessaire d'assurer que leurs régions d'action soient mutuellement exclusives (pas de chevauchements entre les régions d'action) afin d'assurer des actions uniformes sur la région de l'évènement. Du fait que les régions d'action soient circulaires, le chevauchement entre ces régions est inévitable, et donc des effets indésirables peuvent se produire, par exemple, considérons un système d'arroseuses automatisées (actionneurs), avec des capteurs d'humidité. Les arroseuses sont activées quand les lectures de capteur d'humidité vont au-dessous d'un certain seuil. Ici, on préfère que seulement un sous-ensemble minimum d'arroseuses soit activé pour couvrir toute la région de sorte que les ressources globales d'arroseuse (l'eau), et l'énergie soient réduites au maximum.

En traitant ce problème, la solution devrait pouvoir (i) couvrir toute la région d'évènement avec un minimum d'actionneurs, (ii) minimiser la consommation des ressources des actionneurs dues aux actions exécutées, et (iii) minimiser l'énergie consommée par les actionneurs.

## II. Le problème d'exclusion mutuelle

Le problème d'exclusion mutuelle est un problème de gestion de conflits d'accès à des ressources. Les processus qui constituent les systèmes distribués peuvent demander la même ressource en même temps. Si la ressource à besoin d'un accès mutuellement exclusif, donc quelques régulations sont exigées pour y accéder, ceci est le problème d'exclusion mutuelle.

Cependant, dans les WSANs, le problème d'exclusion mutuelle consiste à réduire au maximum les chevauchements entre les régions d'action des actionneurs choisis pour réagir sur un évènement donné.

### III. Classification des algorithmes d'exclusion mutuelle

#### III.1. L'exclusion mutuelle dans les réseaux statiques

Lamport [5] a proposé le premier algorithme d'exclusion mutuelle dans les systèmes répartis, cet algorithme est basé sur les permissions, où chaque processus a besoin d'obtenir la permission de tous les autres processus du système pour entrer dans sa section critique, ce qui nécessite un nombre de messages égale à  $3 \times (n - 1)$ , un autre algorithme basé sur les permissions proposé par Ricart et Agrawala [6] a réduit ce nombre à  $2 \times (n - 1)$  messages.

Maekawa [7] a proposé un schéma dans lequel un processus n'a pas besoin d'obtenir la permission de tous les autres processus du système pour entrer dans sa section critique, ce schéma nécessite un nombre de messages égale à  $3 \times (\sqrt{n} - 1)$  par entrée dans la section critique.

Dans la catégorie des algorithmes basés sur les jetons, Suzuki et Kassami [8] ont proposé un algorithme qui requiert au plus  $n$  messages par entrée dans la section critique, ce nombre est réduit à  $O(\log n)$  dans l'algorithme de Raymond [9] qui utilise une structure logique basée sur une arborescence enracinée.

#### III.2. L'exclusion mutuelle dans les réseaux cellulaires

Badrinath [10] présente un algorithme d'exclusion mutuelle qui s'adapte à un environnement composé de stations de base et des unités mobiles.

L'idée de cet algorithme est de faire un découplage entre la mobilité des sites et la construction des algorithmes pour des environnements mobiles. Pour cela, les algorithmes distribués sont structurés de manière à ce que la majeure partie des communications et du calcul induit par l'algorithme soit prise en charge par le réseau statique. Il suggère que la structure de données qui englobe l'état de l'algorithme réside dans les sites fixes, alors que le rôle des unités mobiles se résume à initier l'exécution de l'algorithme et à recevoir le résultat de l'exécution.

Une autre solution est présentée par Singhal [11] où l'exécution de l'algorithme se fait seulement entre les nœuds demandant l'accès à la section critique.

### III.3. L'exclusion mutuelle dans les réseaux mobiles Ad Hoc

Le problème de l'exclusion mutuelle dans les réseaux Ad Hoc est plus compliqué parce que ces derniers n'utilisent aucune infrastructure fixe, d'où la nécessité de concevoir des algorithmes qui s'adaptent à ce nouvel environnement caractérisé par ses fréquentes déconnexions, changement de topologie et de partitionnement, et qui prend en compte les caractéristiques physiques des unités mobiles (vitesse CPU, bande passante limitée, espace de stockage, source d'énergie limitée). Ils doivent être aussi efficaces, performants et tolérants aux pannes.

Des travaux dans la littérature dans les réseaux adhoc [34,35,36,37] ont été effectués.

### III.4. L'exclusion mutuelle dans les WSANs

Dans les réseaux WSANs, si plusieurs actionneurs sont requis afin d'agir sur un événement donné, il est nécessaire d'assurer que leurs régions d'action soient mutuellement exclusives afin d'assurer des actions uniformes sur la région de l'évènement. Du fait que les régions d'action sont circulaires, le chevauchement entre ces régions est inévitable, et donc des effets indésirables peuvent se produire. Ainsi, selon la nature de l'application, l'action peut avoir plusieurs conséquences, comme :

- (i) La sur-utilisation des ressources d'actionneur, comme un gaspillage de ressource d'eau dans une application de lutte contre le feu,
- (ii) Opération incorrecte : par exemple des capteurs de chaleur et des vibreurs d'alarme (actionneurs) dans une application automatisée d'alarme-de-feu, où il y a une signature unique liée à la fréquence et à la tonalité avec lesquelles l'alarme se déclenche. Ici, quand un feu est détecté, seulement un sous-ensemble minimum de vibreurs non voisins devraient être activés, de sorte que la signature ne devienne pas brouillée et le feu soit détecté, car si deux vibreurs voisins se déclenchent, ils peuvent s'interférer et donc la tonalité n'affirme pas le feu.
- (iii) Une situation catastrophique, par exemple, les actionneurs qui éjectent le gaz de poison dans le cas d'intrus où une dose du gaz paralyse un être humain mais deux doses peuvent le tuer.

Donc la définition classique du problème d'exclusion mutuelle tel connu dans les systèmes distribués et les réseaux adhoc et qui suppose que chaque point dans la région d'intérêt (ou région d'évènement) est couvert par un seul actionneur n'est pas valide pour les WSANs. Vedantham et al.[1] ont défini l'exclusion mutuelle dans les WSANs comme suit: *Etant donné un ensemble d'actionneurs dans une région d'évènement, quel est le sous-ensemble minimum d'actionneurs qui couvre toute la région d'évènement de tel sorte qu'ils existent un minimum de chevauchements entre les régions d'action ?*

La figure 10 présente les différentes notations utilisées par Vedantham et al. [1]. Ces notations vont nous aider à présenter formellement les types d'exclusion mutuelle proposés dans les WSANs.

Variable	Description
$R$	Région d'évènement (surface occupée par l'évènement)
$a_1 \dots a_n$	Ensemble d'actionneurs dans la région $R$ (leurs champs d'action chevauchent avec $R$ )
$M$	Ensemble minimum d'actionneurs pour couvrir la région $R$
$A_i$	Région couverte par l'actionneur $a_i$ dans $R$ (champs d'action de $a_i$ dans $R$ )
$R_M$	Région couverte par l'ensemble $M$ (Union de toutes les régions d'action de l'ensemble $M$ ) dans $R$
$r_i$	Surface dans $R$ couverte par $a_i$ seulement (pas de chevauchement avec des actionneurs dans $M$ )
$n_i$	Surface dans $R$ couverte par $a_i$ et un autre actionneur.
$o_i$	Surface dans $R$ couverte par $a_i$ et 2 autres actionneurs au minimum
$VM_i$	La fonction objective individuelle pour l'actionneur $a_i$

**Figure 10 : Notations d'exclusion mutuelle utilisées par Vedantham et al.**

La figure 11 illustre les différentes régions qu'on peut avoir dans un WSAN selon les notations présentées dans la figure 10.

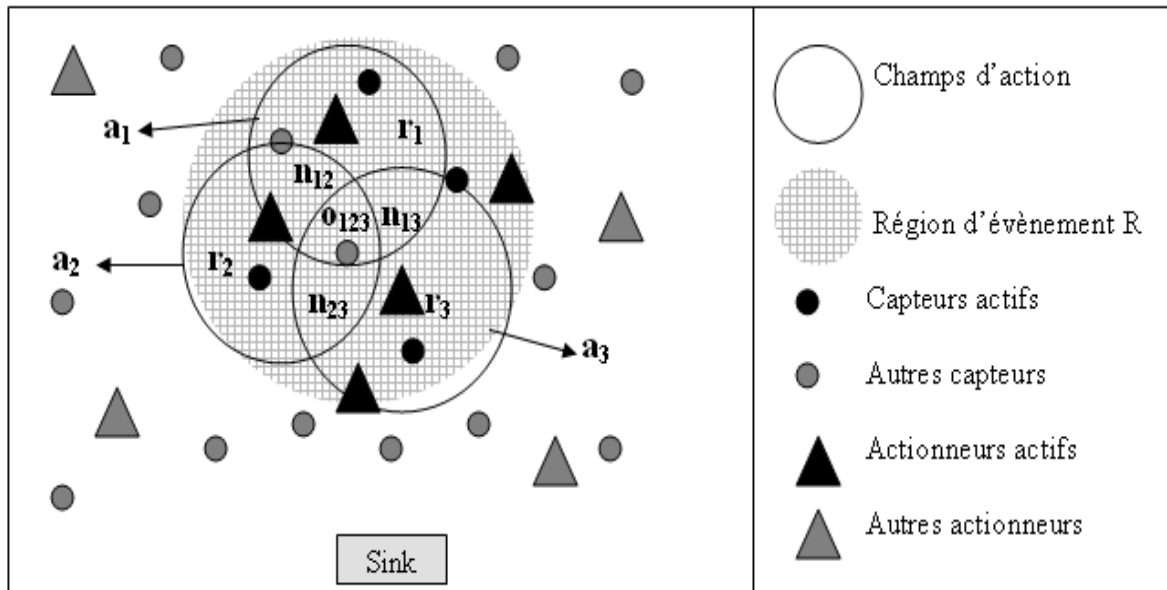


Figure 11 : Différentes régions basées sur la notation

D'après les notations précédentes :

$$o_1 = o_2 = o_3 = o_{123} = A_1 \cap A_2 \cap A_3,$$

$$n_{12} = (A_1 \cap A_2) - o_1, \quad n_{13} = (A_1 \cap A_3) - o_1, \quad \text{et donc } n_1 = n_{13} \cup n_{12}$$

$$r_1 = A_1 - (n_1 \cup o_1).$$

Vedantham et al. [1] présentent quatre types d'exclusion mutuelle dans les WSANS. L'objectif de chaque type est de maximiser une fonction objective totale définie par la somme des fonctions objectives individuelles.

Une fonction objective individuelle d'un actionneur  $a_i$  ( $VM_i$ ) est le bénéfice (valeur non négative) que l'actionneur peut apporter s'il est choisi dans l'ensemble de couverture  $M$ .

### III.4.1. Exclusion mutuelle basée ressource

Le but de ce type d'exclusion mutuelle est de déterminer l'ensemble minimum d'actionneurs  $M$  qui maximise les régions non chevauchées (non-overlapped area) dans la région d'évènement.

En d'autres termes, on doit maximiser la fonction objective globale définie par la somme de différentes fonctions objectives individuelles, où la fonction objective individuelle d'un actionneur est donnée par :  $VM_i = r_i$

Dans la figure 11 : Initialement  $M = \{ \}$

- On remarque que si  $r_3 > r_1$  et  $r_3 > r_2$  alors  $M = \{ a_3 \}$
- si  $r_1 > r_2$  alors  $M = \{ a_3, a_1 \}$
- Il ne reste que  $a_2$  et donc on choisit  $a_2$  et donc  $M = \{ a_3, a_1, a_2 \}$ .

### III.4.2. Exclusion mutuelle basée type de chevauchement

Dans le type précédent, on tient compte seulement des régions non-chevauchées mais ici on tient compte aussi des nouvelles régions chevauchées (régions où il y a un seul chevauchement).

Le but est de maximiser la fonction objective globale définie par la somme de différentes fonctions objectives individuelles, où la fonction objective individuelle d'un actionneur est donnée par :

$$VM_i = r_i - \alpha \times n_i$$

Dans cette équation,  $\alpha$  est une constante qui représente le coût engendré en ayant un chevauchement entre deux régions d'action dans la région d'évènement.

Si on prend la figure 11 : Initialement  $M = \{ \}$ ,

On peut choisir n'importe quel actionneur sans avoir de chevauchements car  $M$  est vide dans ce cas là on ajoute  $a_3$  à  $M$  car il couvre plus la région d'évènement, puis on remarque que les actionneurs restants se chevauchent avec  $a_3$  et donc on calcule la valeur de  $VM_i = r_i - \alpha \times n_i$  (ici on suppose que  $\alpha = 1$ ) pour  $i=1,2$  et on ajoute l'actionneur qui a le  $VM_i$  le plus grand, dans notre cas c'est  $a_1$  puis  $a_2$  (car  $a_1$  génère moins de chevauchement avec  $a_3$  par rapport à  $a_2$ ).

### III.4.3. Exclusion mutuelle basée région de chevauchement

Dans ce type d'exclusion mutuelle, il n'est pas seulement nécessaire de maximiser la surface de la région non chevauchée couverte par chaque actionneur mais aussi également il est important de minimiser la surface des régions chevauchées quelle que soit leur intensité (le nombre de chevauchements n'est pas important).

La fonction objective globale est définie par la somme des différentes fonctions objectives individuelles, où la fonction objective individuelle d'un actionneur est :

$$VM_i = r_i - \beta \times (n_i + o_i)$$

Où,  $\beta$  est une constante qui représente le coût engendré en ayant n'importe quel type de chevauchement dans la région d'évènement  $(n_i, o_i)$ .

#### III.4.4. Exclusion mutuelle basée intensité de chevauchement

L'intensité de chevauchement d'une région désigne le nombre de fois qu'un chevauchement se produit pour une région particulière. L'exclusion mutuelle d'intensité de chevauchement est le cas le plus générique du problème d'exclusion mutuelle, où les actionneurs sont choisis non seulement basés sur les nouveaux et les anciens chevauchements mais aussi sur l'intensité des chevauchements.

La fonction objective individuelle d'un actionneur est donnée par :

$$VM_i = r_i - \sum_{I_i} \beta_{I_i} \times I_i \times (n_i, o_i)$$

Où  $\beta_{I_i}$  est le coût engendré en ayant un chevauchement avec intensité  $I_i$  dans la région d'évènement.

Quelque soit l'algorithme d'exclusion mutuelle dans les WSANs, il doit garantir deux buts :

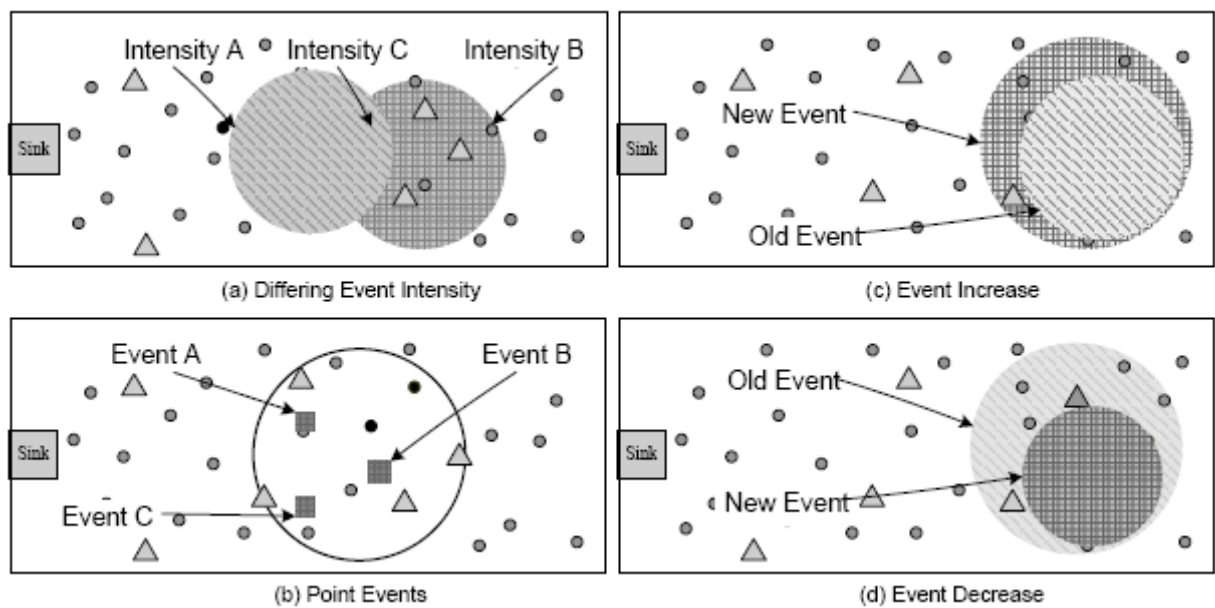
- Overhead : le cout engendré par l'exclusion mutuelle doit être faible c'est à dire l'approche doit être capable de traiter le problème d'exclusion mutuelle avec un minimum de cout de communication.
- Couverture : l'approche doit impérativement assurer la couverture totale de la région de l'évènement.

#### IV. Autres variantes d'événements

Dans notre projet, nous traitons le problème d'exclusion mutuelle en considérant seulement des événements qui se produisent dans des régions statiques et qui nécessite seulement un cycle d'exécution. Cependant, pour d'autres types d'applications, les évènements peuvent prendre différentes formes et intensités [1] :

• *Différentes Intensités d'événement* : Dans quelques applications, l'intensité de l'événement peut ne pas être la même à travers la région d'événement (voir la figure 12 (a)). Si l'intensité d'événement est différente dans deux régions non chevauchées, alors les actions devraient être effectuées pour refléter l'intensité désirée de l'événement. Dans ce cas, l'application peut exiger des cycles multiples de l'action dans certaines régions selon l'intensité d'évènement.

Considérons l'application de lutte contre le feu, où l'intensité du feu est différente dans différentes régions. Ici, il est nécessaire que les actions effectuées soient équivalentes à l'intensité d'évènement dans cette région.



**Figure 12 : variantes d'évènements**

• *Événements point/Multi-point* : Jusqu'ici, on a seulement décrit le problème d'exclusion mutuelle pour un événement régional. Le problème pour les événements point/multi-points est tout à fait différent. Dans ce cas, la définition consiste à minimiser le nombre d'actionneurs qui traitent tous les événements points dans le réseau (voir la figure 12 (b)). Par exemple, considérons une application de détection d'intrus, où les capteurs captent l'image et les actionneurs éjectent un gaz toxique. Dans ce cas, l'événement est l'occurrence d'intrus, qui est un événement point/multi-points. Ainsi, le problème d'exclusion mutuelle consiste à réduire au

minimum les actionneurs qui couvrent tous ces événements points basés sur la fonction objective.

- *Événement dynamique*: Ce défi se produit quand il y a des ronds multiples d'opération pour traiter un événement. Quand la région d'événement augmente, il est nécessaire que l'exclusion mutuelle soit fournie pour la nouvelle région d'événement comme représenté sur la figure 12 (c). Dans l'application d'extincteur, supposons qu'il y a initialement un feu dans une région et si le feu se propage dans la région d'évènement avec le temps, l'approche d'exclusion mutuelle devrait s'adapter pour traiter le problème pour la nouvelle région d'évènement. De même, quand l'évènement diminue, il est nécessaire que l'approche d'exclusion mutuelle soit seulement exécutée dans la région d'évènement courante (voir 12 (d)).

Dans l'exemple d'extincteur, si l'intensité du feu est maximale au centre de la région, il est certain qu'on éteigne d'abord le feu dans l'extrémité de la région d'évènement. Après ce temps, l'exclusion mutuelle est exigée seulement dans la nouvelle région d'évènement.

## V. Conclusion

Dans ce chapitre, nous avons en premier lieu introduit un état de l'art sur le problème d'exclusion mutuelle dans les différents systèmes (réseaux statiques, réseaux cellulaires, réseaux mobiles Ad Hoc), nous avons montré que le problème d'exclusion mutuelle dans les WSANs prend une autre définition qui diffère de celle décrite dans les autres réseaux. Nous avons introduit des exemples pratiques qui illustrent le problème d'exclusion mutuelle dans les WSANs. Vedantham et al. [1] ont proposé quatre types d'exclusion mutuelle qui diffèrent selon le besoin de l'application, de plus ils ont proposé les différentes formes d'évènement qui peuvent exister; dans le cadre de notre projet nous avons considéré le cas d'un évènement régional, statique, et nécessite une seule action à la fois.

## I. Introduction

Le problème d'exclusion mutuelle dans les WSN n'a pas reçu une grande attention dans la littérature. Vedantham et al. [1] sont les seuls qui ont traité ce problème. Ils ont proposé deux approches pour le résoudre, une centralisée et une autre distribuée. Cependant, il existe des travaux en liaison avec ce problème qui visent généralement à garantir une couverture de la région de l'évènement, rappelons que l'exclusion mutuelle dans les WSNs requière une couverture totale de la région de l'évènement auquel on rajoute d'autres contraintes liées à la minimisation des ressources des actionneurs.

## II. La solution de Vedantham et al.

Vedantham et al. [1], ont proposé deux approches : une centralisée et une autre distribuée. Nous présentons dans ce qui suit les deux approches, ensuite nous donnons une petite synthèse qui va introduire notre solution.

### II.1. Approche centralisée

#### II.1.1. Description de l'approche

Dans cette approche, quand les capteurs détectent un évènement, ils envoient un message au sink pour l'informer; le sink, en connaissant la topologie globale du réseau, construit l'ensemble d'actionneurs de couverture, en sélectionnant à chaque étape l'actionneur avec la fonction objective maximale où la fonction objective est définie par le type d'exclusion mutuelle (voir la section III.4 dans le chapitre 2), jusqu'à ce que l'ensemble sélectionné  $M$  couvre toute la surface de l'évènement (voir l'algorithme 1). Ensuite, il transmet une commande à chaque actionneur de l'ensemble  $M$  pour exécuter l'action requise.

Soit  $M$  l'ensemble d'actionneurs de couverture. Initialement,  $M$  est un ensemble vide. L'algorithme commence par inclure un actionneur arbitraire qui se situe complètement dans la région (maximise la fonction objective). À chaque étape, l'algorithme choisit un actionneur,  $MAX\_ACTOR$  qui a la fonction objective maximale,  $VM_{MAX\_ACTOR}$  (lignes 14-17 dans l'algorithme 1). L'actionneur est ajouté à l'ensemble de couverture d'actionneur,  $M$ , et la valeur de  $R_M$  est mise à jour en

ajoutant la surface nouvellement couverte de cet actionneur, MAX\_BENEFIT. En outre, l'ensemble d'actionneurs restant, est mis à jour en enlevant cet actionneur (lignes 19-21 dans l'algorithme 1). N'importe quel actionneur,  $a_i$ , qui a une région de chevauchement avec cet actionneur, met à jour la région non couverte,  $r_i$ , les régions chevauchées,  $n_i$ , et  $o_i$  (lignes 22-24 dans l'algorithme 1). L'algorithme se termine quand la région d'évènement est contenue dans  $R_M$  (ligne 10 dans l'algorithme 1).

```

Input
1    $a_1, a_2, \dots, a_n$  : actionneurs dans le WSN, S : Sink
2   C : Commande envoyée à la région d'intérêt
3    $R_C$  : Région d'intérêt de la commande
4   : Ensemble d'actionneurs dont leurs champs d'action se chevauchent avec  $R_C$ 
5    $R_i$  : Champ d'action de l'actionneur  $a_i$ 

Output
6   L'ensemble minimal d'actionneurs de couverture de  $R_C$ 

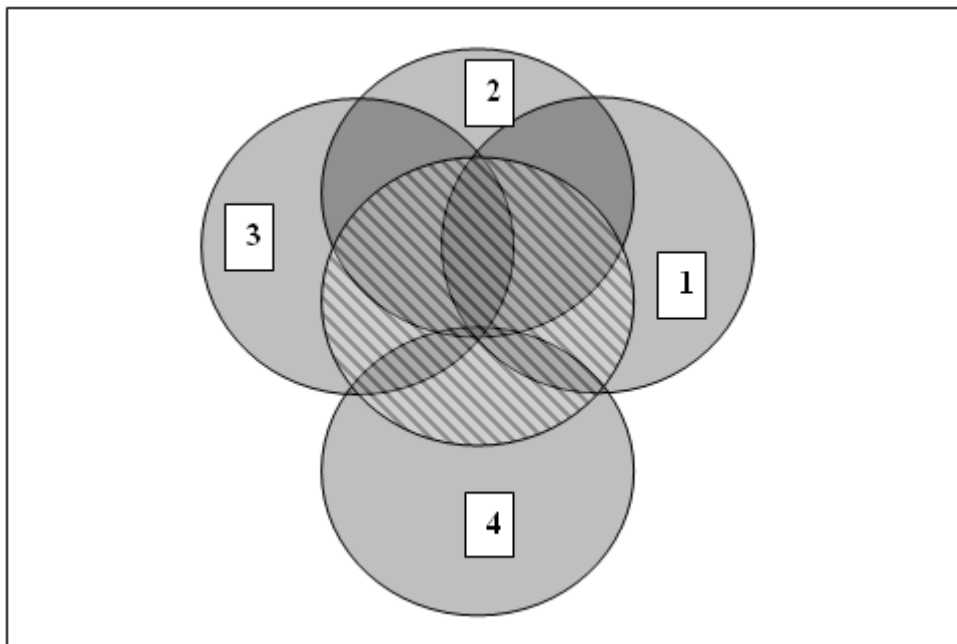
Calculer l'ensemble minimal d'actionneurs de couverture de la région  $R_C$ 
7   M : L'ensemble d'actionneurs sélectionnés comme membre de l'ensemble de couverture à
   chaque étape donnée
8    $R_M$  : Région couverte par M
9   w =
10  while ( $R_C \not\subset R_M$ )
11      MAX_BENEFIT = 0 (ou  $-\alpha \times \pi \times R^2$ )
12      for each  $a_i \in w$ 
13           $VM_i$  = fonction de bénéfice de  $a_i$ 
14          if ( $VM_i > MAX\_BENEFIT$ )
15              MAX_BENEFIT =  $r_i$ 
16              MAX_ACTOR = i
17          end if
18      end for
19       $M = M \cup MAX\_ACTOR$ 
20       $R_M = R_M + MAX\_BENEFIT$ 
21       $w = w - MAX\_ACTOR$ 
22      for each ( $a_i \in w$ ) and ( $(R_i \cap R_{MAX\_ACTOR}) \neq \emptyset$ )
23          Update  $r_i, o_i, n_i$ 
24      end for
25  end while
26  return M

```

**Algorithme 1 : Algorithme centralisé de Vedantham et al.**

### II.1.2. Exemple d'illustration

Dans la figure 13, le cercle raillé représente la région d'évènement, les cercles 1, 2, 3, 4 représentent les régions d'action des actionneurs a1, a2, a3, a4 respectivement. Supposons que le type d'exclusion mutuelle choisi est de minimiser tout chevauchement possible (celui décrit dans la section III.4.3 dans le chapitre 2). Initialement  $M=\emptyset$ , dans ce cas, n'importe quelle actionneur choisi n'a pas d'intersection avec M, donc on choisit celui qui couvre le plus la région d'évènement (c'est à dire celui qui à le plus grand  $r_i$ ); dans ce cas c'est a2.



**Figure 13 : Exemple d'illustration de l'approche centralisée de Vedentham**

$M = \{a2\}$ . Maintenant on remarque que tous les actionneurs qui restent se chevauchent avec M. Dans ce cas là on calcule pour chaque actionneur restant sa valeur de  $VM_i$  qui vaut :  $VM_i = r_i - \times (n_i + o_i)$  ; et supposons qu'on donne un poids plus grand pour les chevauchements (c'est à dire on encourage le choix des actionneurs qui ont un minimum de chevauchements avec M) par exemple  $=2$ . Dans ce cas là, il est clair que c'est a4 qui sera ajouté à M.

$M=\{a2,a4\}$ . De la même manière on ajoute a1 et a3 pour couvrir la région d'évènement.

## II.2. Approche distribuée

### II.2.1. Définition

Vedantham et al. [1] ont aussi proposé une approche distribuée et complètement localisée pour traiter le problème d'exclusion mutuelle dans les réseaux WSANs.

L'approche est basée sur la notion de la région de dépendance où :

La région de dépendance d'un capteur est donnée par la région avec un rayon égale à la somme du rayon de capture et du rayon d'action, tandis que celle d'un actionneur est donnée par la région avec un rayon égale à deux fois le rayon d'action.

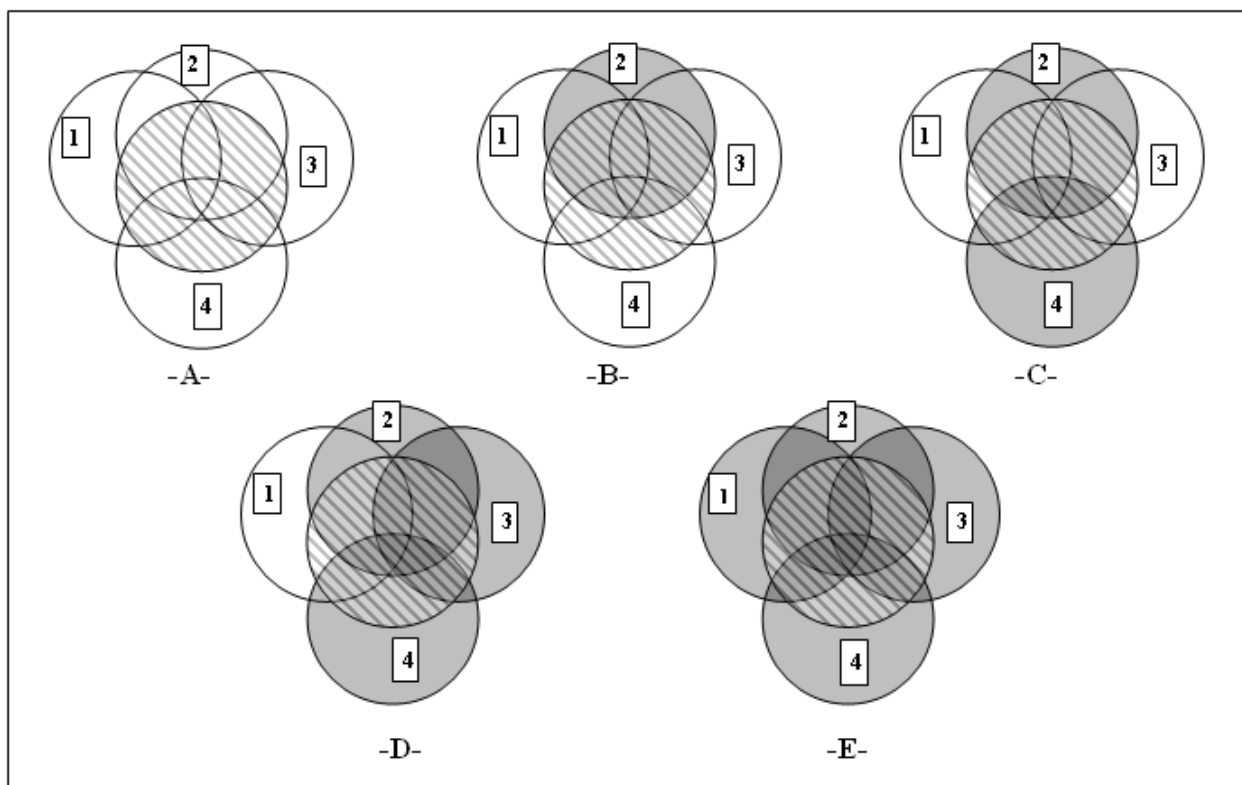
### II.2.2. Description de l'approche

Dans l'approche distribuée, quand le capteur détecte un évènement, il envoie un message REQUEST à chaque actionneur dans sa région de dépendance. Ensuite, un processus de coordination est lancé. Dans ce processus, chaque actionneur attend une durée de temps avant d'envoyer un message NOTIFY à chaque actionneur dans sa région de dépendance. Ce message contient la valeur du bénéfice que l'actionneur va apporter s'il est sélectionné. Chaque actionneur met à jour sa valeur de bénéfice après la réception d'une valeur de bénéfice d'un autre actionneur. Le temps d'attente pour chaque actionneur est inversement proportionnel à sa fonction objective (bénéfice). De cette manière, les actionneurs avec la meilleure valeur de bénéfice seront les premiers à être choisis pour agir sur la région d'évènement.

## III. Etude de la solution de Vedantham et al.

Contrairement à l'approche centralisée, l'approche distribuée à l'inconvénient de consommer plus de messages et de répondre plus rapidement à l'évènement. Cependant, comme chaque actionneur prend des décisions autonomes et ne peut pas avoir l'état précis de tout le réseau, le coût de l'ensemble de couverture construit par l'approche distribuée sera moins optimal que celui de l'approche centralisée. D'un autre côté, l'approche centralisée est moins performante en termes de nombre de messages générés et le délai entre la détection de l'évènement et l'action.

Les approches proposées sont des heuristiques et ne sont pas des algorithmes qui donnent des résultats optimaux en terme d'exclusion mutuelle car le fait de choisir à chaque fois l'actionneur avec la fonction objective maximale n'implique pas que l'ensemble des membres choisis est optimal. Ce point est bien illustré dans la figure 14, où: le cercle raillé représente la région d'évènement, les cercles 1, 2, 3, 4 représentent respectivement les champs d'actions des actionneurs  $a_1$ ,  $a_2$ ,  $a_3$ ,  $a_4$  respectivement. Supposons que le type d'exclusion mutuelle choisi est de minimiser tout chevauchement possible (celui décrit dans le paragraphe III.4.3. dans le deuxième chapitre).



**Figure 14 : Exemple d'illustration de l'optimalité de la solution**

Dans ce cas, l'algorithme centralisé (vue dans le paragraphe II.1.) va choisir les actionneurs suivants dans l'ordre :  $a_2$  (car il couvre plus la région d'évènement), ensuite on choisit l'actionneur  $a_4$  car il génère moins d'intersection avec  $a_2$ , ensuite on choisit l'actionneur  $a_3$ , puis enfin  $a_1$ . Avec l'algorithme centralisé, on remarque que la solution génère 4 intersections entre deux actionneurs ( $a_1$  avec  $a_2$ ,  $a_2$  avec  $a_3$ ,  $a_3$  avec  $a_4$ , et  $a_1$  avec  $a_4$ ) et deux intersections entre 3 actionneurs  $\{a_1, a_2, \text{ et } a_3\}$  et  $\{a_1, a_3, \text{ et } a_4\}$  (voir la figure 14), mais cette solution malgré qu'elle assure une couverture totale

de la région d'évènement elle n'est pas la meilleure : en effet, la solution qui est intuitive et plus optimale que celle-ci consiste à choisir seulement  $a_1$  et  $a_3$  et  $a_4$ . Dans ce cas, on couvre toute la région d'évènement et on génère moins de chevauchements (il existe 3 chevauchements seulement).

Il a été montré dans [12,13,14] que le calcul de la couverture optimale d'un ensemble donné pour un graphe de réseau donné (ce qui est ressemblé à notre problème) est NP-Difficile. Donc le choix des actionneurs va se faire selon une méthode approximative (heuristique).

#### **IV. Travaux liés au problème d'exclusion mutuelle dans les WSANs**

Durant notre recherche dans la littérature, nous avons trouvé des travaux qui ne portent pas le même thème que celui que nous étudions, mais nous avons trouvé dans ces travaux des astuces qui peuvent aider à réduire le problème d'exclusion mutuelle dans les WSANs, nous présentons un petit aperçu sur ces travaux un par un et nous tirons dans la conclusion les points qui nous semblent utiles.

##### **IV.1. La solution de Tommaso Melodia et al.**

Tommaso Melodia et al. [15] ont proposé une solution pour résoudre le problème de coordination dans les WSANs, on parle de la coordination capteur-actionneur et de la coordination actionneur-actionneur. Concernant la première phase, ils ont divisé la région de l'évènement en plusieurs partitions (*clusters*), dans chaque partition il y a un actionneur qui est chargé de collecter les informations de l'évènement à partir des capteurs sous sa responsabilité (voir la figure 15). On ce qui concerne la deuxième phase de coordination, ils ont proposé une technique qui permet de réduire l'énergie nécessaire pour les actions déclenchés qui consiste à réduire le rayon d'action de quelques actionneurs et cela basé sur une fonction qui cherche à maximiser la moyenne de l'énergie résiduelle dans l'ensemble des actionneurs de chaque partition. De cette manière, ils devisent la charge d'action sur plusieurs actionneurs et ils considèrent l'énergie présente sur chaque actionneur, ce qui va engendrer une longue durée de vie des actionneurs.

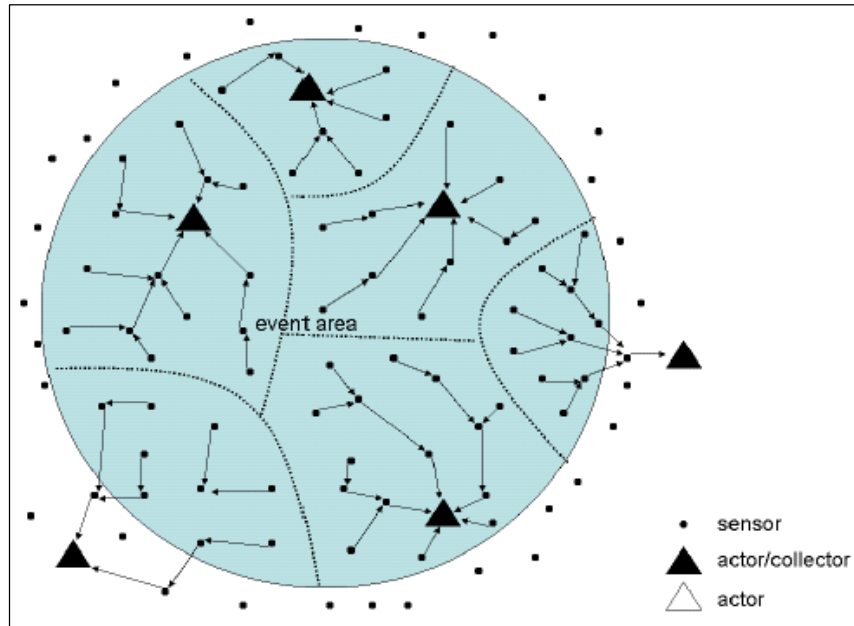


Figure 15 : Partitionnement dans la région de l'évènement

Dans la figure 16, on voit que les régions d'action des actionneurs sont différentes et dans ce cas la on voit clairement que la réduction de la région d'action permet de minimiser l'énergie d'action.

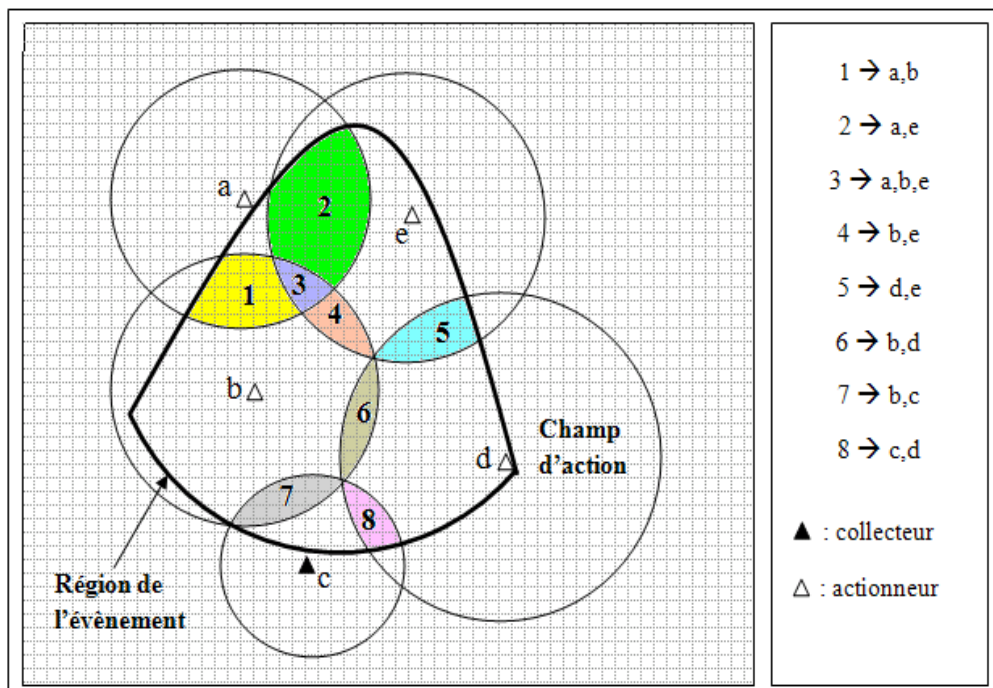


Figure 16 : Régions chevauchées et non-chevauchées pour un collecteur c

## IV.2. La solution de Akkaya et Younis

L'un des plus grand problèmes dans les différentes applications des WSNs est de pouvoir fournir les services en n'importe qu'elle point dans le réseau. Ceci est dénommé le problème de couverture dans les WSNs. Cependant, ces services doivent être effectués rapidement afin de fournir les effets désirés. La solution à ce problème doit attribuer des emplacements spéciaux pour les actionneurs dans la région d'intérêt afin de répondre le plus rapidement possible à l'évènement détecté.

Maximiser la couverture des actionneurs et minimiser le délai de transmission des données sont deux objectives contradictoires, d'où la nécessité d'avoir un compromis entre les deux objectives. Dans la pratique, la distribution des capteurs n'est pas uniforme sur la région d'intérêt. D'où le placement des actionneurs pour une couverture maximale peut conduire à une topologie de réseau inefficace et/ou étend les chemins des données capteurs-actionneurs (par exemple placer des actionneurs là où il n'y a pas de capteurs).

### IV.2.1. Définition du problème

Akkaya et Younis [16] ont défini leur problème comme suit : « Etant donnée une région d'intérêt connue et un ensemble de capteurs et d'actionneurs initialement déployé aléatoirement dans la région d'intérêt, quelle est la meilleure localisation pour les actionneurs qui peut fournir une couverture maximale de la région d'intérêt et minimise le temps de collection de données et le temps d'exécution de l'action des actionneurs en considérant les contraintes d'énergie des capteurs.

### IV.2.2. Description de la solution

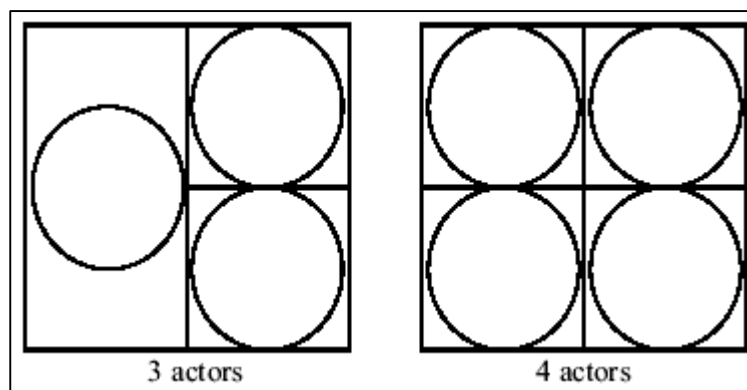
Akkaya et Younis [16] ont proposée COLA, « a COverage and Latency aware Actor placement scheme for WSNs » : est un algorithme de placement des actionneurs qui assure la couverture et réduit le temps de collection des données dans les WSNs.

Leur solution est exécutée en deux phases :

### 1) Construction du réseau :

La première phase inclût le positionnement initial des actionneurs et le groupement des capteurs en clusters.

**1.1) Placement initial des actionneurs :** le problème de couverture peut être achevé soit d'une manière statique ou dynamique [28]. La couverture statique est le problème de déploiement des actionneurs dans une configuration statique, de tel sort que chaque point dans la région est sous la couverture d'un actionneur à chaque instant. La couverture dynamique est effectuée par un algorithme qui explore et couvre la région avec des parties constantes [28]. Dans COLA, l'emplacement initial des actionneurs est effectué en divisant la région en plusieurs cellules égales dont leurs nombre correspond au nombre d'actionneurs, puis ils positionnent les actionneurs au centre de ces cellules. La figure 17 illustre cette idée, où le cercle représente le champ de communication d'un actionneur (région où l'actionneur peut communiquer avec une autre entité du réseau). Un tel placement fournit une couverture maximale en prévenant un chevauchement entre les champs de communication des actionneurs.



**Figure 17 : Redéploiement des actionneurs**

Afin de calculer ces emplacements, un algorithme distribué est utilisée :

- 1- Chaque actionneur diffuse son id et sa localisation.
- 2- En utilisant les ids, le nombre d'actionneurs, l'emplacement de chaque actionneur et les limites de la région d'évènement, chaque actionneur exécute un algorithme récursif pour diviser la région en cellules et

sauvegarde l'emplacement des centres de ces cellules dans un arbre binaire.

- 3- Chaque actionneur calcule la cellule la plus proche de lui, ensuite il se déplace vers l'emplacement optimal.

**1.2) La formation des clusters :** Après le positionnement des actionneurs, l'étape suivante est de grouper les capteurs dans plusieurs clusters différents; les actionneurs commencent à découvrir leurs voisinages de capteurs, et d'une manière collaborative entre eux ils déterminent ce que l'on appelle « *clusters* ». Pour chaque cluster, un actionneur est désigné d'être le « *cluster-head* » qui sera chargé de rassembler et traiter les données à partir des capteurs de son cluster.

## 2) Relocation des actionneurs pour minimiser la latence

Afin de réduire le temps nécessaire pour la collection de données, les auteurs proposent une technique qui va déplacer les actionneurs de leurs emplacements courant vers un emplacement qui va réduire les chemins capteurs-actionneurs.

## IV.3. La solution de McLaughlan et Akkaya

Le problème de couverture considéré dans ce papier a été étudié dans la littérature dans le contexte des systèmes multi-robots. Le problème de couverture est défini d'être la maximisation de la région totale couverte par les actionneurs. Ceci peut être effectué d'une manière statique ou dynamique [29]. Si la région considérée est grande et le nombre de robots est limité, seule la couverture dynamique peut être considéré [30]. Dans ce cas, les robots se déplacent continuellement de sorte à couvrir la région désirée.

### IV.3.1. Définition du problème

McLaughlan et Akkaya [17] définit le problème comme suit : « Etant donnée un ensemble de capteurs placé initialement d'une manière aléatoire dans la région d'intérêt et un nombre d'actionneurs disponible, on cherche à placer les actionneurs et

partitionner le réseau en clusters tel que chaque cluster-head est un actionneur et que la couverture totale des actionneurs est maximisée ».

### IV.3.2. Description de la solution

McLaughlan et Akkaya [17] ont proposé un algorithme qui s'exécute en deux phases, la première cherche à trouver l'emplacement des actionneurs, tandis que la deuxième cherche à faire partitionner le réseau.

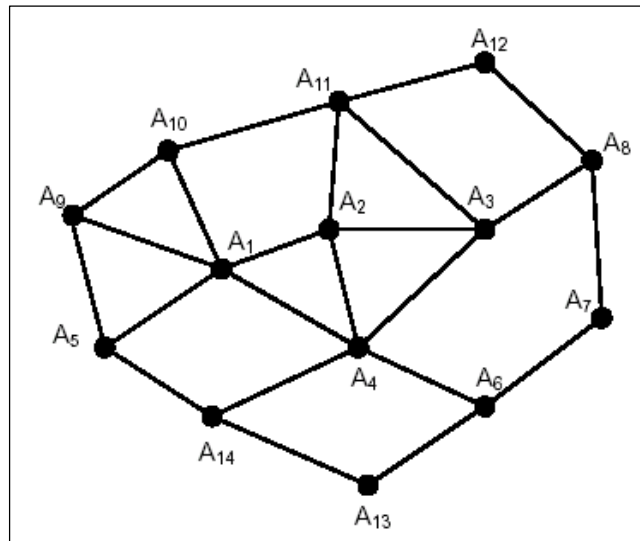
**1- Placement des actionneurs :** le but est de disperser les actionneurs d'une façon uniforme dans la région d'intérêt de tel sorte que le chevauchement entre les régions d'action soit minimal. Dans cette approche, le réseau de capteurs est modélisé comme étant un graphe  $G=(V,E)$  où  $V$  est l'ensemble des nœuds et  $E$  est l'ensemble des arcs. Dans ce graphe, deux nœuds connectés par un arc s'ils peuvent communiquer entre eux.

Soient les notations suivantes :

- a. Deux éléments dans  $V$  sont dits voisins s'ils peuvent communiquer entre eux de façon directe.
- b. Independent Set (IS) : est un sous ensemble de  $V$  tel qu'il n'existe pas deux nœuds dans l'ensemble qui sont voisins dans  $V$ .
- c. Dominating Set (DS) : soit  $S$  un DS. Alors  $S$  est un sous ensemble de  $V$  tel que chaque nœud dans  $V - DS$  est voisin au maximum à un nœud dans  $S$ . En d'autre terme on peut atteindre toutes les nœuds dans  $V - DS$  à partir d'au moins un élément de  $S$ .
- d.  $k$ -Dominating Set ( $k$ -DS) : soit  $S$  un  $k - DS$ . Alors  $S$  est un sous ensemble de  $V$  tel que chaque nœud dans  $V - S$  peut être atteint au minimum à un nœud dans  $S$  en  $k$ -saut au maximum.
- e. Independent Dominating Set (IDS) : est un DS tel qu'il y a pas deux dominateurs voisins dans cet ensemble. C'est-à-dire que chaque dominateur n'est pas atteint par un seul saut depuis un autre dominateur.

f. **k-Independent Dominating Set (k-IDS)** : est un k-DS tel que chaque deux dominateurs peuvent communiquer entre eux à partir de k-sauts et plus.

La figure 18 fournit des exemples de : DS, IDS, k-DS et k-IDS dans une topologie de réseau donnée pour les actionneurs :  $A_0, A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8, A_9, A_{10}, A_{11}, A_{12}, A_{13}, A_{14}$ .



**Figure 18** : dans ces 14 nœuds, il y a : 1)  $A_2, A_4, A_8, A_9, A_{13}$  qui forment un DS. 2)  $A_4, A_8, A_9, A_{11}, A_{13}$  qui forment un IDS. 3)  $A_2$  et  $A_4$  qui forment un 2-DS. 4)  $A_1$  et  $A_7$  qui forment un 2-IDS. 5)  $A_4$  qui forme un 3-DS.

Une bonne couverture peut être achevée en déterminant l'ensemble IS car on aura peu de chevauchements entre les champs des actionneurs; en déterminant l'ensemble DS on aura la couverture maximale, d'où l'ensemble IDS va garantir la couverture de tous les capteurs en minimisant les chevauchements entre les champs d'action. Cependant, pour réduire le nombre de partitions, on doit minimiser le nombre d'éléments dans l'IDS d'où la convergence vers l'ensemble k-IDS ce qui est un problème NP-Hard.

**2- Couverture basée-clustering** : après avoir formé l'ensemble, k-IDS l'étape qui suit est d'associer les capteurs à leurs clusters-head.

## V. Conclusion

Dans ce chapitre, nous avons introduit la solution proposée par Vedantham et al. [1], ces derniers ont proposé deux approches pour résoudre le problème d'exclusion

mutuelle dans les WSANs une centralisée et l'autre distribuée. Cependant leur objective était de maximiser les régions d'action non chevauchées pour tous les actionneurs qui se trouvent dans la région d'évènement, afin d'utiliser le minimum des ressources de ces actionneurs. Leur algorithme fonctionne en sélectionnant, à chaque étape l'actionneur qui couvre la plus grande surface de la région d'évènement non couverte encore. Il se termine quand l'ensemble d'actionneurs de couverture couvre toute la surface de l'évènement. De cette manière les auteurs supposent intuitivement que le fait de maximiser les régions d'action non chevauchées va conduire à minimiser le taux de ressources supplémentaire. Cependant cette supposition est fournie sans preuve. De plus ils considèrent que le coût de la région chevauchée, couverte par deux actionneurs ou plus, dépend seulement de la taille de sa région. Comme chaque nouveau chevauchement produit une quantité supplémentaire de ressources d'actionneur, cette méthode ne peut pas estimer correctement le coût supplémentaire des ressources des actionneurs.

Nous avons vu que le choix d'une approche (centralisée et distribuée) peut s'avérer non pertinent vu que chaque approche possède des inconvénients et des avantages, le choix va dépendre certainement de l'application, de plus nous avons vu que le problème d'exclusion mutuelle dans les WSANs est un problème NP-Complexe et donc il n'existe pas une solution qui trouve le sous ensemble de couverture le plus optimal, d'où la nécessité de trouver une approche approximative. De plus dans les autres travaux cités par la suite, on a vu que les champs d'action des actionneurs peuvent être réduits dans certaines applications afin de réduire les ressources des actionneurs, et nous avons vu également que notre problème peut être ressemblé au problème de couverture qui utilise la mobilité des actionneurs et/ou le bon choix des actionneurs, mais avec des contraintes en plus. Dans notre solution, nous donnons une autre manière pour calculer l'ensemble couverture, et nous introduisons trois alternatives qui vont nous permettre de minimiser le taux de chevauchement entre les champs d'action de l'ensemble de couverture à savoir : optimiser le nombre d'actionneurs, réduire le rayon d'action des actionneurs, et la mobilité des actionneurs.

## I. Introduction

Dans le chapitre précédent, nous avons montré qu'il n'existe pas de solution qui donne le sous-ensemble optimal de couverture, mais il existe des solutions approximatives à ce problème. Les travaux réalisés dans [15,16,17] utilisent la mobilité des actionneurs ou la réduction du champs d'action afin d'optimiser la région couverte par les actionneurs. Dans [16,17], il a été proposé des solutions au problème de couverture dans les WSANs qui ressemble à notre problème traité, cependant la couverture considérée est une couverture qui essaye d'assurer des délais de communications plus courts et non pas l'optimisation du coût de l'action.

La solution proposée par Vedenham et al. [1], sélectionne les actionneurs selon une fonction objective qui représente généralement le coût dû au chevauchement entre les champs d'action. Cependant, dans notre solution nous proposons une autre alternative pour le calcul de la fonction objective. Notre solution va comporter deux phases :

(1) phase d'initialisation : qui va chercher l'ensemble de couverture initial basé sur notre fonction objective.

(2) phase d'optimisation : qui va introduire certaines améliorations à l'ensemble de couverture initial.

Dans ce chapitre, nous allons présenter notre modèle réseau, notre définition du problème d'exclusion mutuelle ainsi que notre algorithme d'exclusion mutuelle basée ressource, appelé Centralized Resource-based Mutual Exclusion Algorithm (CRMEA).

## II. Modèle du réseau

- Nous considérons un modèle de réseau WSAN dans lequel les capteurs, le sink, et les actionneurs sont aléatoirement distribués d'une manière à couvrir la surface du WSAN (région d'intérêt).
- Nous supposons qu'on connaît au préalable la localisation de chaque actionneur grâce à un algorithme de localisation [18] et il existe un protocole de routage fiable pour délivrer les messages à n'importe quel actionneur ou capteur [19].
- Champs d'action ou région d'action d'un actionneur  $a_i$ , dénoté par  $A_i$ , est la région circulaire sur laquelle  $a_i$  peut agir.

- Notre modèle supporte quatre types d'actionneurs : (1) statique avec un champ d'action fixe, (2) statique avec un champ d'action dynamique, (3) mobile avec un champ d'action fixe, et (4) mobile avec un champ d'action dynamique.
- Les évènements sont modélisés par des disques de rayon dénoté  $R_e$ . Ce disque est appelé région d'évènement.
- Nous considérons les évènements statiques (leur région d'évènement ne change pas durant l'exécution de l'action) et qui nécessitent un seul rang d'exécution.

### III. Formulation du problème

Avant de définir le problème d'exclusion mutuelle :

- Un point dans une région d'évènement  $R$  est dit de nouvelles notions comme  $K$ -actionneur-couvert. Aussi, les symboles :  $\mathcal{K}$ ,  $A$ , ...etc écrits aussi en italique ( $K \geq 1$ ) s'il se trouve dans la région d'action de  $K$  actionneurs actifs.
- Une région  $A$  est dite  $K$ -actionneur-couverte ( $K \geq 1$ ) si chaque point dans  $A$  est  $K$ -actionneur-couvert.  $K$  est appelé le degré de couverture de  $A$ , dénoté par  $\mathcal{K}(A)$ .
- Etant donnée une région d'évènement  $R$ , et un ensemble d'actionneurs actifs dont le champ d'action se chevauche avec  $R$ , dénoté par  $\mathcal{A}$ , le système  $\mathcal{C} = (R, \mathcal{A})$  est appelé la configuration de couverture d'actionneur si chaque point de  $R$  est dans le champ d'action d'au moins un actionneur dans  $\mathcal{A}$ . Une configuration  $\mathcal{C}$  crée un ensemble de sous-régions disjointes dans  $R$ . Cet ensemble est dénoté par  $\mathcal{P}$  (voir la figure 19).
- Nous définissons  $S_i = \{q \in \mathcal{P} \mid \mathcal{K}(q) = i\}$  comme étant l'ensemble des sous régions dans la région d'évènement  $R$  dont le degré de couverture est  $i$ .
- Le degré maximal de couverture d'une configuration  $\mathcal{C}$ , dénoté par  $\mathcal{K}(\mathcal{C})$ , est le plus grand degré de couverture parmi toutes les sous régions de la région d'évènement  $R$ .
- Considérons un actionneur  $a_i$  qui couvre une région  $A \subseteq R$  dont la taille est  $Size(A)$ , la quantité de ressources dépensées par  $a_i$  sur  $A$  est  $Res(A, a_i) = \alpha_i \times Size(A) \times T_i$ , tel que  $T$  est le temps requis par  $a_i$  pour exécuter l'action et  $\alpha_i$  est la quantité des ressources consommées / unité de temps / unité de

surface. Pour des raisons de simplicité, nous ignorons  $T_i$  et  $T_i$  dans le reste du mémoire. La quantité de ressources dépensées par une configuration  $\xi = (R, \Omega)$  sur la région d'évènement R est défini par :

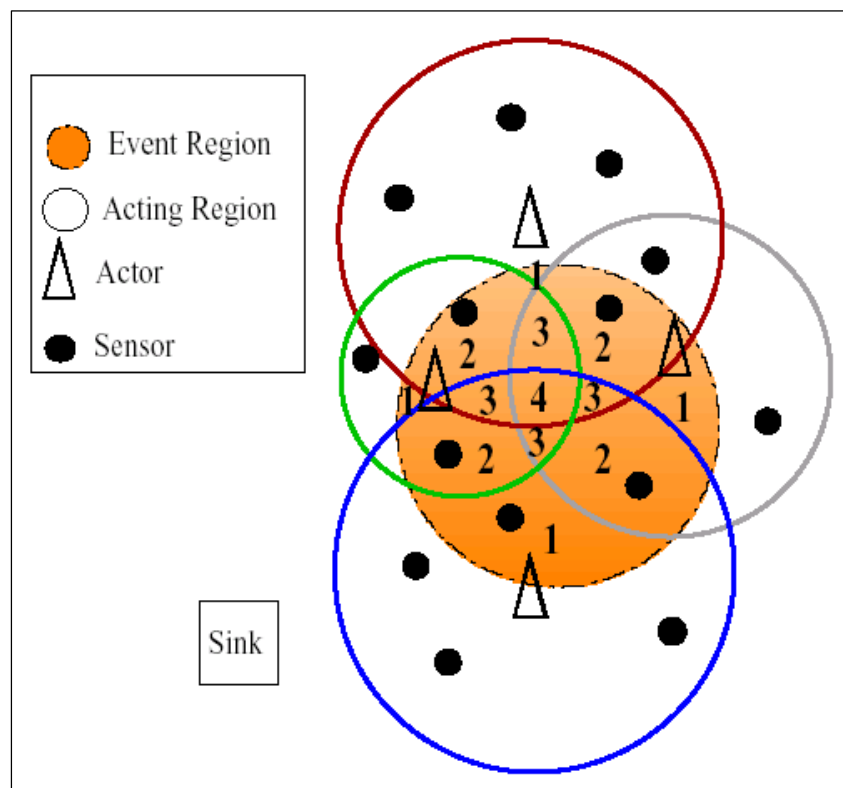
$$\begin{aligned}
 \text{Res}(\xi) &= \sum_{a_i \in \Omega} \text{Size}(A_i \cap R) \text{ elle peut être écrite aussi : } \sum_{i=1}^{\Delta(\xi)} \sum_{a \in S_i} i \times \text{Size}(a) \\
 \text{Res}(\xi) &= \sum_{i=1}^{\Delta(\xi)} \sum_{a \in S_i} i \times \text{Size}(a) = \sum_{a \in S_1} \text{Size}(a) + \sum_{i=2}^{\Delta(\xi)} \sum_{a \in S_i} (i-1) \times \text{Size}(a) \\
 &= \sum_{a \in S_1} \text{Size}(a) + \sum_{a \in S_2} \text{Size}(a) + \sum_{a \in S_2} \text{Size}(a) + \sum_{a \in S_3} \text{Size}(a) + \sum_{a \in S_3} 2 \times \text{Size}(a) + \dots \\
 &\quad \sum_{a \in S_{\Delta(\xi)}} \text{Size}(a) + \sum_{a \in S_{\Delta(\xi)}} (\Delta(\xi) - 1) \times \text{Size}(a) \\
 &= \text{Size}(R) + \sum_{i=2}^{\Delta(\xi)} \sum_{a \in S_i} (i-1) \times \text{Size}(a)
 \end{aligned}$$

- Nous pouvons remarquer que Res( ) est composé de deux parties : (1) Size(R), qui est exactement la quantité de ressource requise pour la région d'évènement R, et (2) l'autre partie qui représente la quantité supplémentaire de ressources dépensées par .
- Le système  $M = (R_M, M)$  est appelé configuration partielle de si :  $R_M \subseteq R$  et chaque point dans  $R_M$  est dans le champs d'action d'au moins un actionneur dans  $M \subseteq \mathcal{A}$ .  $P^M$  est l'ensemble des sous régions disjointes dans  $R_M$  par  $M$ .
- Nous définissons  $M = \{q \in P^M \mid (q)=i\}$  comme l'ensemble des sous régions dans la région d'évènement  $R_M$  dont le degré de couverture est i.
- Le problème d'exclusion mutuelle que nous définissons consiste à trouver l'ensemble d'actionneurs  $M \subseteq \mathcal{A}$  tel que : la région couverte par M est  $R_M = R$  et la quantité de ressources d'actionneurs supplémentaire est minimale.

Formellement, notre objectif est de minimiser la fonction objective suivante :

$$f^M = \text{Min} \left( \sum_{i=2}^{\Delta(\xi^M)} \sum_{a \in S_i^M} (i-1) \times \text{Size}(a) \right)$$

La figure 19 représente un exemple sur la configuration de couverture proposée dans notre travail, les valeurs 1, 2, 3 et 4 dans la figure correspondent au degré de couverture de la sous région (i de la formule précédente). Les disques blancs représentent les champs d'action des actionneurs et celui en orange représente la région de l'évènement. Les triangles sont des actionneurs et les petits disques noirs sont des capteurs.



**Figure 19 : La configuration de couverture des actionneurs**

#### IV. Algorithme d'Exclusion Mutuelle Centralisé basée-Ressource (Centralised based-Ressource Mutual Exclusion Algorithme : CRMEA)

Quand les nœuds capteurs détectent l'existence d'un nouvel évènement, ils envoient un message signalant l'évènement au sink. Basé sur ces messages, nous supposons que le sink est capable de déterminer les coordonnées et le rayon de l'évènement. Le sink exécute notre algorithme d'exclusion mutuelle (CRMEA) en deux phases : (1) la phase initiale qui calcule l'ensemble d'actionneurs  $M$  qui minimise la quantité supplémentaire de ressources  $f^M$ , et (2) la phase d'optimisation qui elle-même constituée des phases suivantes : (a) réduction du nombre d'actionneurs, (b) réduction du champ d'action et/ou mobilité des actionneurs, comme indiqué sur la figure 20.

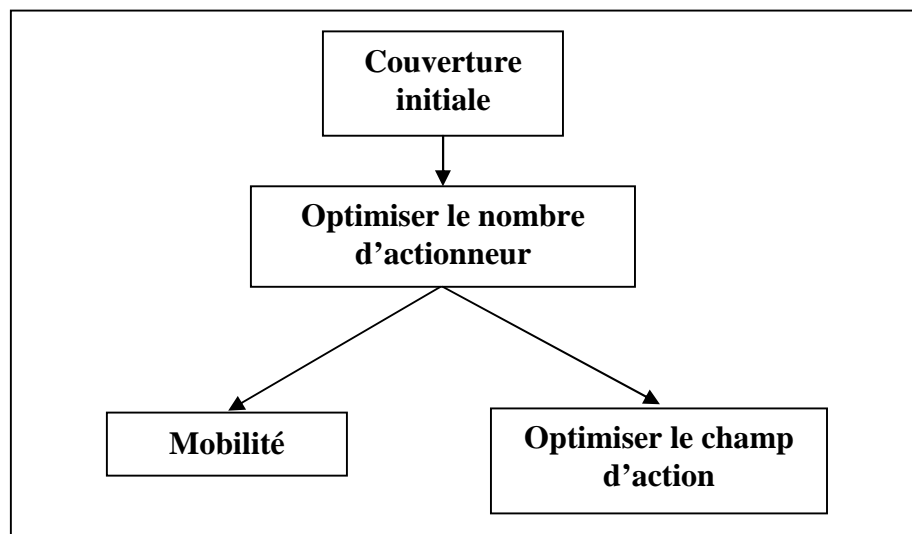


Figure 20 : L'organigramme général de CRMEA

##### IV.1. La phase d'initialisation de CRMEA

Dans cette phase, nous cherchons à obtenir une couverture initiale de la région d'évènement. Comme il est décrit dans l'algorithme 2, le nœud sink choisit à chaque étape l'actionneur non encore sélectionné qui soit : il ne génère aucune ressource supplémentaire (sa sélection ne va pas créer un nouveau chevauchement dans la région d'évènement) (lignes 14-16 dans l'algorithme 2), soit il génère un minimum de ressources supplémentaires (lignes 9-12 dans l'algorithme 2). Nous pouvons constater

que le premier actionneur sélectionné par le sink ne génère aucune ressource supplémentaire. L'actionneur sélectionné est ajouté à  $M$  (l'ensemble d'actionneur de couverture), la surface couverte  $R_M$  est mise à jour, et l'actionneur sélectionné est supprimé de l'ensemble des actionneurs non encore sélectionnés  $w$  (lignes 17-19 dans l'algorithme 2). L'algorithme se termine quand la région d'évènement  $R$  est totalement couverte par  $M$  (ligne 5 dans l'algorithme 2).

```

1 :  $M = \emptyset$ 
2 :  $R_M = \emptyset$ 
3 :  $w =$ 
4 :  $f^M(R_M, M) = 0$ 
5 : while  $(R \not\subset R_M)$  and  $(R \cap R_M) \neq \emptyset$  do
6 :    $\Delta_{\min} = \text{INFINITY}$ 
7 :   for each  $a_i \in w$  do
8 :      $\Delta C = f^{M \cup a_i} - f^M$ 
9 :     if  $\Delta C < \Delta_{\min}$  then
10 :        $\Delta_{\min} = \Delta C$ 
11 :       selected =  $a_i$ 
12 :     end if
13 :   end for
14 :   if  $\Delta_{\min} = 0$  then
15 :     selected = {  $a_i \in w \mid (A_i \cap R)$  est le plus grand }
16 :   end if
17 :    $M = M \cup \text{selected}$ 
18 :    $R_M = R_M \cup (A_i \cap R)$ 
19 :    $w = w - \text{selected}$ 
20 : end while
21 : return  $M$ 

```

**Algorithme 2 : La phase d'initialisation de CRMEA**

## IV.2. La phase d'optimisation du nombre d'actionneur dans la région d'évènement

La première étape peut générer des actionneurs qui sont redondants, pour cela dans cette phase nous allons chercher dans l'ensemble  $M$ , les actionneurs dont la suppression n'affecte pas la couverture totale de la région d'évènement. Il se peut qu'on ait plusieurs actionneurs qui peuvent être supprimés. Dans ce cas, on commence par celui qui minimise au maximum la fonction objective  $f^M$  (lignes 6-16 dans

l'algorithme 3) jusqu'à ce qu'il n'y ait plus d'actionneur à supprimer c'est-à-dire il n'existe pas d'actionneurs supplémentaires (ligne 2 dans l'algorithme 3).

```

1 : Bool = TRUE
2 : while Bool = TRUE do
3 :   Bool = False
4 :   max = 0
5 :   selected = ∅
6 :   for each  $a_i \in M$  {  $M$  est l'ensemble envoyé par le sink } do
7 :     if  $(M - a_i)$  couvre encore  $R$  then
8 :       Bool = TRUE
9 :        $\Delta C = f^M - f^{M-a_i}$ 
10 :      if  $C > \max$  then
11 :        max = C
12 :        selected =  $a_i$ 
13 :      end if
14 :    end if
15 :  end for
16 :   $M = M - \text{selected}$ 
17 : end while
18 : return M

```

### Algorithme 3 : La phase d'optimisation du nombre d'actionneurs

#### IV.3. La phase d'optimisation du champ d'action dans la région d'évènement

Cette phase peut être exécutée seulement quand les actionneurs sont capables de changer leurs champs d'action. Selon [3], les champs d'action (rayons d'action) des actionneurs peuvent être contrôlés.

Dans notre approche, pour chaque actionneur  $a_i \in M$ , le sink essaye de trouver s'il est possible de réduire le rayon d'action de  $a_i$  tout en assurant une couverture totale de la région d'évènement  $R$ . Pour cela, il exécute la fonction «ReduceRadius» afin d'obtenir le nouveau rayon d'action de l'actionneur (ligne 6 dans l'algorithme 4). Si la fonction retourne -1, cela veut dire que le rayon d'action ne peut pas être réduit encore parce que s'il réduit sa valeur alors la région d'évènement  $R_e$  ne va pas être totalement couverte et donc cet actionneur n'est pas intéressant et doit être ignoré (lignes 7-8 dans l'algorithme 4). Le sink commence par les actionneurs dont la minimisation de leurs rayon d'action minimise au maximum la fonction objective (lignes 5-16 dans

l'algorithme 4), et continue de faire ainsi jusqu'à ce qu'il n'y ait plus de champ d'action à réduire (ligne 2 dans l'algorithme 4).

```

1 : w = M { M est l'ensemble retourné par l'algorithme 3 }
2 : while w ≠ ∅ do
3 :   selected = ∅
4 :   min = fM
5 :   for each ai ∈ w do
6 :     Rnew = ReduceRadius (ai)
7 :     if Rnew = -1 then
8 :       w = w - ai
9 :     else
10 :      calculer fnew, la nouvelle fonction objective assumant que ARi = Rnew
11 :      if fnew < min then
12 :        min = fnew
13 :        selected = ai
14 :      end if
15 :    end if
16 :  end for
17 :  w = w - selected
18 :  acting rang sélectionné = Rnew
19 : end while

```

#### Algorithme 4 : La phase d'optimisation du champ d'action dans R

La figure 21(c) montre la nouvelle configuration obtenue après l'exécution de l'algorithme 4 sur la configuration de la figure 21(a). Le nouveau champ d'action d'un actionneur  $a_i$ , dénoté par  $R_{new}$ , doit être le maximum entre les deux valeurs suivantes :

(1) La distance entre  $a_i$  et les points d'intersection du champ d'action de chaque voisin de  $a_i$  avec la région d'évènement. Ces points d'intersection doivent appartenir à la sous région  $(A_i \cap R)$  (exemple : points b et d dans les figure 21(b)). L'ensemble de ces points est dénoté par  $U(a_i)$ .

(2) La distance entre  $a_i$  et le point d'intersection entre chaque deux voisins de  $a_i$ . Les points d'intersection doivent appartenir à la sous région  $(A_i \cap R)$  (exemple : le point c dans la figure 21(b)). L'ensemble de ces points d'intersection est dénoté par  $V(a_i)$ .

Comme il est montré sur la figure 21(b), le nouveau rayon de l'actionneur  $a_1$  est le  $\max([l_1, b], [l_1, c], [l_1, d])$ ; où  $l_1$  est la position de l'actionneur  $a_1$ .

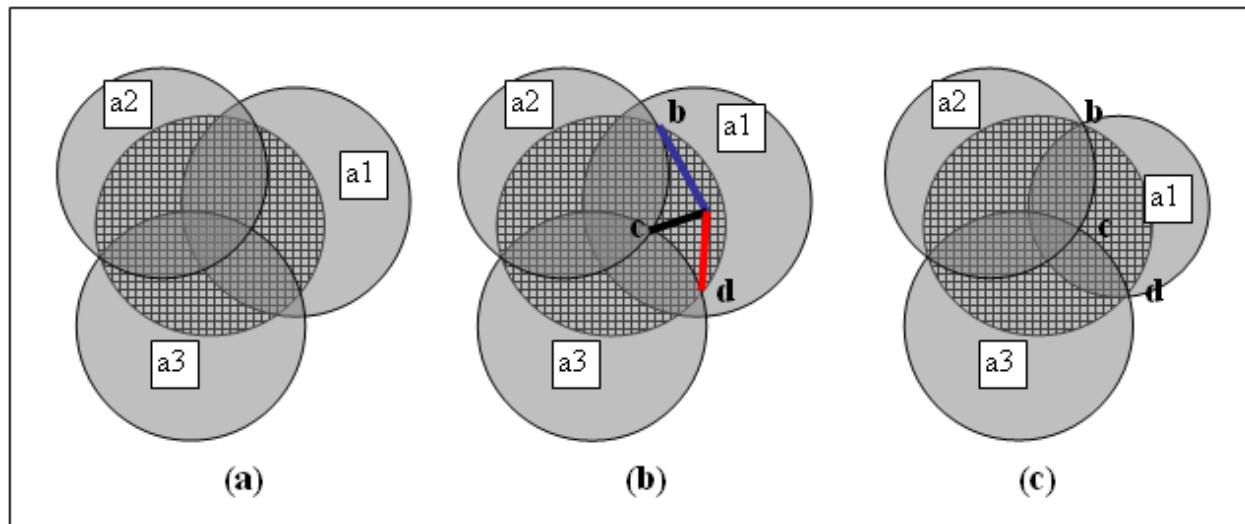


Figure 21 : Exemple d'exécution de l'optimisation du champ d'action

#### IV.4. La phase d'optimisation par mobilité dans la région d'évènement

Cette phase peut être exécutée seulement quand les actionneurs sont mobiles. Dans notre approche, pour chaque actionneur  $a_i \in M$ , le sink essaye de trouver s'il est possible de déplacer l'actionneur loin de R tout en préservant la couverture de la région d'évènement R. Pour faire cela, il exécute la fonction « Move » pour obtenir les nouvelles coordonnées de l'actionneur  $a_i$  (ligne 6 dans l'algorithme 5). Si la fonction retourne (-1,-1) cela veut dire que l'actionneur ne peut pas être déplacé de sa position actuelle parce que s'il se déplace, la région d'évènement R ne sera pas couverte, et donc l'actionneur sera ignoré (lignes 7-8 dans l'algorithme 5). Le sink commence avec les actionneurs dont leur mouvement minimise au maximum la fonction objective (lignes 5-16 dans l'algorithme 5), et il continue cette procédure jusqu'à qu'il n'y ait plus d'actionneur à déplacer (ligne 2 dans l'algorithme 5).

```

1 : w = M { M est l'ensemble retourné par l'algorithme 3 }
2 : while w ≠ ∅ do
3 :   selected = ∅
4 :   min = fM
5 :   for each ai ∈ w do
6 :     (newX,newY) = Move(ai)
7 :     if (newX,newY) = (-1,-1) then
8 :       w = w - ai
9 :     else
10 :      calculer fnew, la nouvelle fonction objective assumant que li=(newX,newY)
11 :      if fnew < min then
12 :        min = fnew
13 :        selected = ai
14 :      end if
15 :    end if
16 :  end for
17 :  w = w - selected
18 :  location sélectionné = (newX,newY)
19 : end while

```

### Algorithme 5 : La phase d'optimisation par mobilité dans R

La figure 22(c) montre la nouvelle configuration obtenue en exécutant l'algorithme 5 sur la configuration de la figure 22(a) passant par l'étape illustrée à la figure 22(b).

L'actionneur dans sa nouvelle position doit couvrir les points  $U(a_i)$  et les points de  $V(a_i)$  (comme on vut dans la phase précédente). Les nouvelles coordonnées de  $a_i$  sont obtenues en effectuant une translation géométrique de sa région d'action. L'actionneur sera déplacé suivant le vecteur  $\vec{l_i m}$  tels que  $l_i$  est la position de  $a_i$  et  $m$  est le centre de la droite  $[b,c]$  ( $b,c$  sont des points  $\in U(a_i)$ ). Dans la figure 22(c), on voit clairement que  $l_i b = l_i c$  rayon de  $a_i$ .

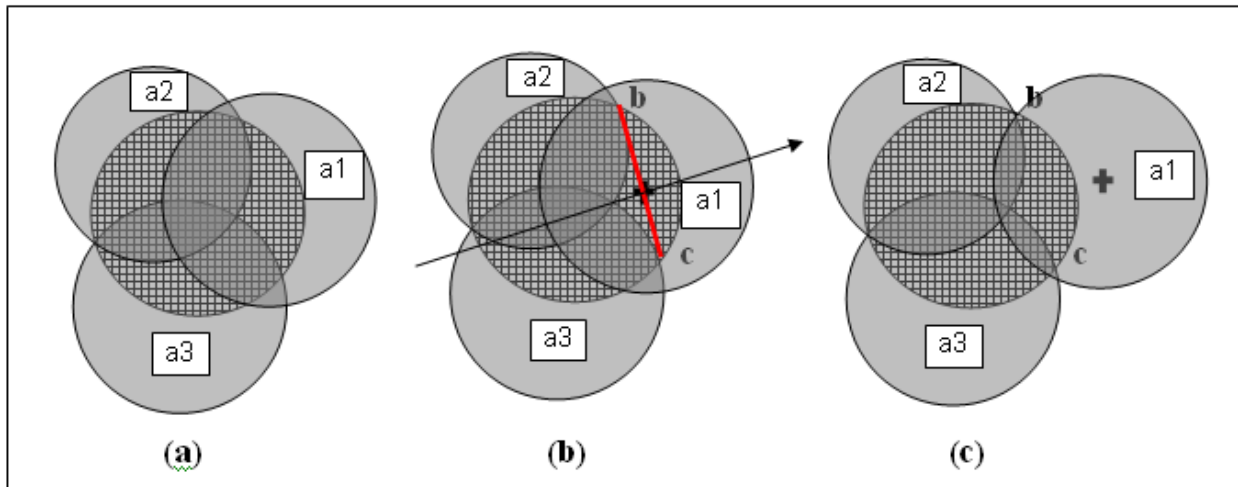


Figure 22 : Exemple d'exécution de l'optimisation par mobilité

#### IV.5. Combinaison de l'optimisation du champ d'action et la mobilité

Les deux extensions d'optimisation proposées dans les algorithmes 4 et 5 peuvent être exécutées une après l'autre pour minimiser au maximum la fonction objective. Dans cette extension, si la taille de l'évènement est inférieure au champ d'action, un actionneur est sélectionné pour se déplacer vers le centre de l'évènement et le couvrir avec un rayon d'action égal au rayon de la région d'évènement.

## V. Conclusion

Dans ce chapitre nous avons présenté notre algorithme d'exclusion mutuelle basée ressource, appelé Centralized Resource-based Mutual Exclusion Algorithm (CRMEA) qui est composée de deux étapes : étape d'initialisation suivie de l'étape d'optimisation, dans chaque étape de notre solution, le choix des actionneurs est basée sur la fonction objective globale qui compte la quantité de ressources supplémentaires dépensées par l'ensemble de couverture M.

Après l'étape d'initialisation, nous exécutons trois phases d'optimisation : la première cherche à supprimer les actionneurs redondants, la seconde cherche à minimiser le rayon d'action de quelques actionneurs dans la région d'évènement, et la phase d'optimisation par mobilité cherche à déplacer quelques actionneurs dans la région d'évènement.

L'objectif de chaque étape d'optimisation est de réduire la quantité de ressources supplémentaires. Dans le prochain chapitre, nous allons valider notre solution par des tests de simulation.

## I. Introduction

Dans ce chapitre, nous allons introduire quelques métriques pour évaluer la performance de notre solution. Puis nous donnons les résultats de nos simulations en fonction de la taille de la région d'évènement et en fonction de la vitesse des actionneurs.

## II. Métriques de performance

Afin d'évaluer l'optimalité de l'ensemble d'actionneurs de couverture construit, nous comparons la performance de l'algorithme centralisé de Vedantham et al. [1] dénoté par VA, avec les différentes versions de CRMEA, qui sont :

- L'algorithme2 (CRMEA initial) suivi par la phase d'optimisation du nombre d'actionneurs : dénoté CRMEA.
- CRMEA suivi par la phase d'optimisation du champ d'action : dénoté par CRMEA+ R.
- CRMEA suivi par la phase d'optimisation par mobilité : dénoté par CRMEA+ M.
- CRMEA + R suivi par la phase d'optimisation par mobilité : dénoté par CRMEA + R + M.
- CRMEA + M suivi par la phase d'optimisation du champ d'action : dénoté par CRMEA + M + R.

La figure 23 décrit les différentes stratégies considérées selon les notations précédentes :

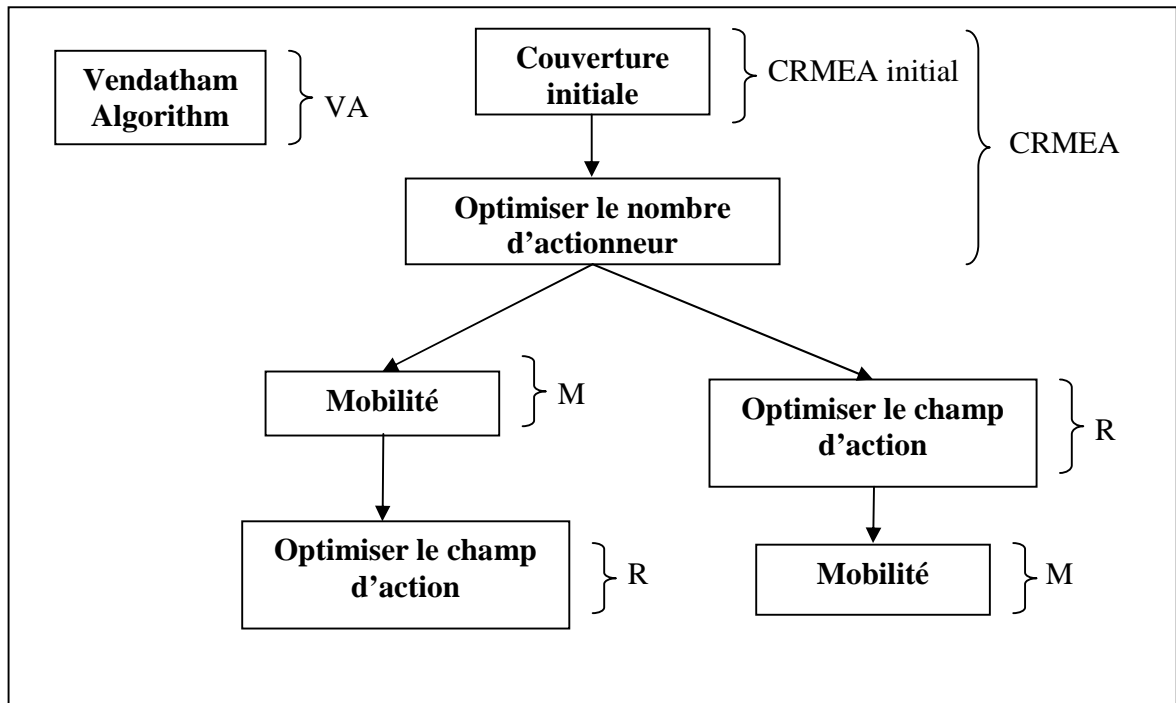


Figure 23 : Notations des stratégies de calcul de couverture

Dans notre simulation, nous avons 120 actionneurs avec un champ d'action de rayon égal à 30 m aléatoirement déployés de façon à couvrir la région d'intérêt qui est de 400 m × 400 m. Dans nos résultats expérimentaux, chaque valeur représente la moyenne de 50 exécutions (une exécution correspond à un évènement donné). Notre application est développée en C sur un PC P4 de 1.8 GHz avec 512mo de RAM.

Nous définissons les métriques suivantes pour évaluer les algorithmes d'exclusion mutuelle :

**1) Coût d'action :** est la quantité de ressources dépensées sur la région d'évènement, elle représente la somme des ressources requises et les ressources supplémentaires.

**2) Coût d'énergie :** l'énergie consommée par tous les actionneurs pour effectuer l'action. Nous supposons qu'un actionneur dont le rayon d'action est R, consomme  $\pi \times R^2$  Joules. Durant le mouvement, l'actionneur consommera une énergie supplémentaire proportionnelle à sa vitesse V.

**3) Degré d'exclusion mutuelle :** c'est le rapport entre le coût d'action et la taille de la région d'évènement donc plus cette valeur est proche de *un (1)* mieux est l'exclusion mutuelle car :

Degré d'exclusion mutuelle =  $(S + f^M(R_M, M)) / S = 1 + (f^M(R_M, M)) / S$  où  $S$  est la surface de la région d'évènement.

**4) Taux d'optimisation du coût :** est défini d'être *un (1)* moins le rapport entre le coût d'action de la stratégie d'optimisation et celui de CRMEA initial suivi de l'optimisation du nombre d'actionneur. Plus cette valeur est proche de 100% plus elle est considérée meilleure.

**5) Degré maximal de couverture :** représente le nombre maximum d'actionneurs qui peuvent couvrir n'importe quel point dans la région d'évènement.

**6) Délai de mobilité :** est le temps requis par l'actionneur pour se déplacer de sa position actuelle vers la position où il effectue l'action.

**7) Energie de mobilité :** est l'énergie due au déplacement des actionneurs.

## IV. Résultats

### IV.1. Coût d'action

La figure 24 représente la variation du coût d'action en fonction du rayon de la région d'évènement pour quatre stratégies : VA, CRMEA, CRMEA + R + M, et CRMEA + M + R.

On remarque que quand le rayon d'évènement augmente, le coût d'action augmente pour les quatre stratégies.

-Pour les rayons 10m et 20m, VA suit CRMEA et CRMEA + R + M suit CRMEA + M + R en terme de coût. Pour le rayon 30m, CRMEA est meilleur que VA alors que CRMEA + R + M et CRMEA + M + R ont les mêmes résultats. Cela peut être expliqué par le fait que pour des rayons d'évènement 30 (rayon des actionneurs) la phase de mobilité constate qu'un seul actionneur suffit pour couvrir la région d'évènement ; d'où CRMEA + R + M et CRMEA + M + R donnent le même coût d'action.

-Pour les rayons 30m, CRMEA est meilleur que VA et CRMEA + R + M est meilleurs que les autres stratégies.

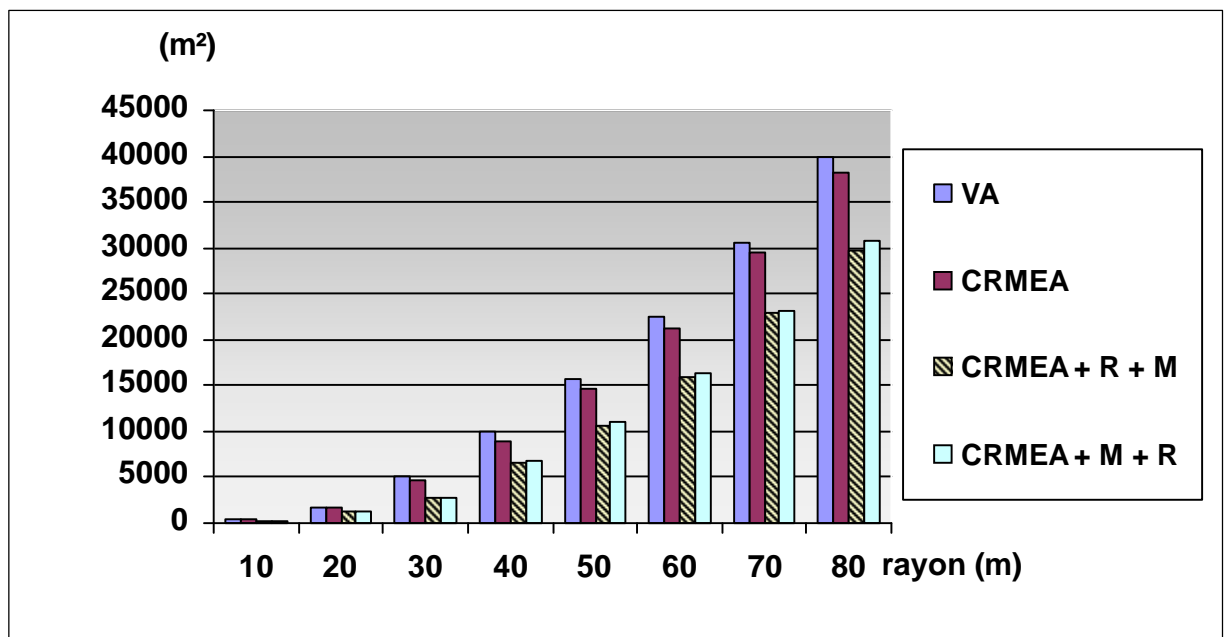


Figure 24 : Le coût d'action en fonction du rayon d'évènement

#### IV.2. Coût d'énergie

La figure 25 représente la variation du coût d'énergie qui inclut l'énergie due à l'action et l'énergie due à la mobilité en fonction des rayons de la région d'évènement pour quatre stratégies : VA, CRMEA, CRMEA + R + M, et CRMEA + M + R.

On remarque que plus le rayon d'évènement augmente le coût d'énergie augmente pour les quatre stratégies.

-Pour les rayons 10m, 20m, et 30m VA et CRMEA donnent des résultats presque similaires car la première utilise une fonction objective qui cherche à maximiser le ri se

qui en résulte un minimum nombre d'actionneur, tandis que CRMEA + R + M et CRMEA + M + R donne toujours le même résultat.

-Pour les rayons 40m CRMEA est meilleur que VA et CRMEA + M + R est meilleur que toutes les autres stratégies.

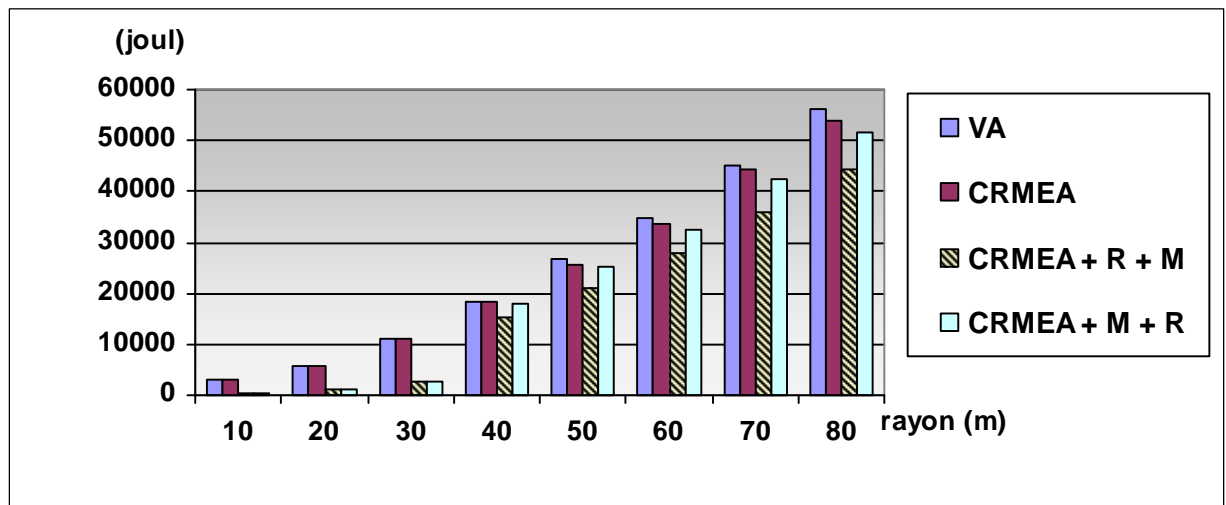
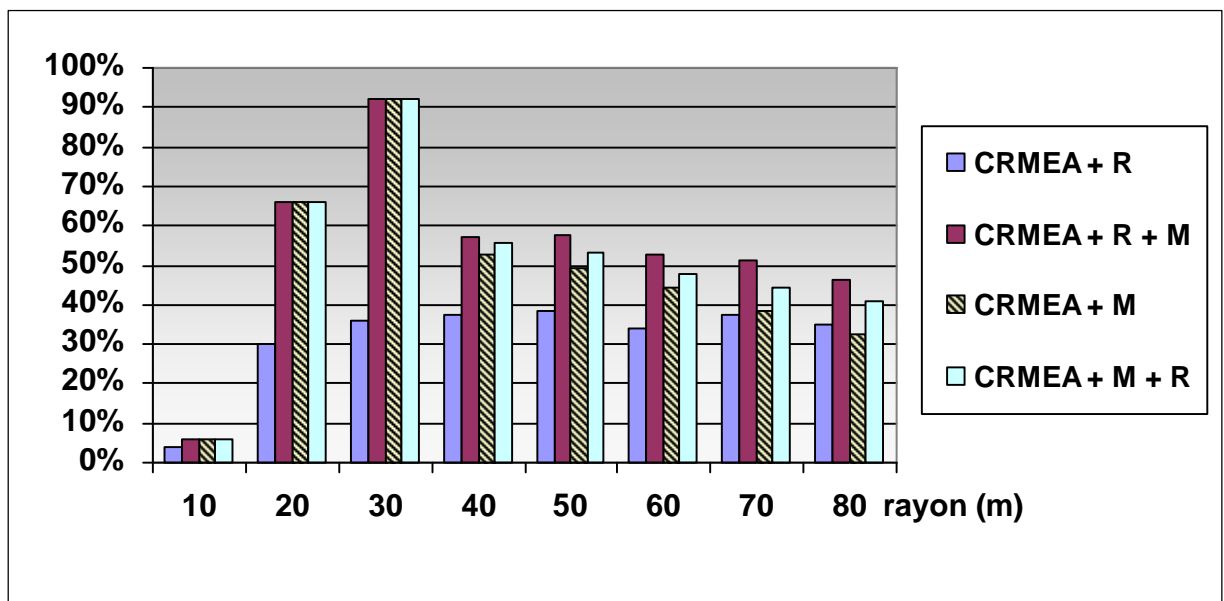


Figure 25 : Le coût d'énergie en fonction du rayon d'évènement

### IV.3. Taux d'optimisation du coût

La figure 25 représente le taux de réduction de la fonction objective par rapport au résultat de CRMEA en fonction des rayons de la région d'évènement pour les quatre stratégies : CRMEA + R, CRMEA + R + M, CRMEA + M, et CRMEA + M + R.



**Figure 26 : Taux d'optimisation du coût en fonction du rayon d'évènement**

On remarque que le taux d'optimisation du coût diminue en fonction du rayon d'évènement, cela est justifié par le fait que la densité des actionneurs est faible dans les extrémités de la région d'intérêt est donc pour les grands évènements on va choisir des actionneurs qui ont un petit taux de chevauchement.

- Pour les rayons 30m, la phase de mobilité constate qu'un seule actionneur peut couvrir la région d'évènement car le rayon d'action = 30m, donc elle choisit un seul actionneur pour le déplacer vers le centre de l'évènement, et pour cela nous remarquons que les stratégies CRMEA + R + M, CRMEA + M, et CRMEA + M + R donne le même taux.

- Pour les rayons 40m, les résultats indiquent le classement suivant : CRMEA + R + M est meilleur que CRMEA + M + R, et ce dernier est meilleur que CRMEA + R, et CRMEA + M classé le dernier.

#### IV.4. Degré d'exclusion mutuelle

La figure 27 représente l'impact de l'augmentation du rayon d'évènement sur le degré d'exclusion mutuelle pour les quatre stratégies : VA, CRMEA, CRMEA + R + M, et CRMEA + M + R.

On remarque que le degré d'exclusion mutuelle augmente en augmentant la taille de l'évènement.

- Pour les rayons 30 m CRMEA + R + M et CRMEA + M + R donnent le même degré car la mobilité fait en sorte à choisir un actionneur pour couvrir la région d'évènement.
- Pour les rayons 40m CRMEA + R + M est la meilleur ensuite vient CRMEA + M + R suivi de CRMEA et en fin VA.

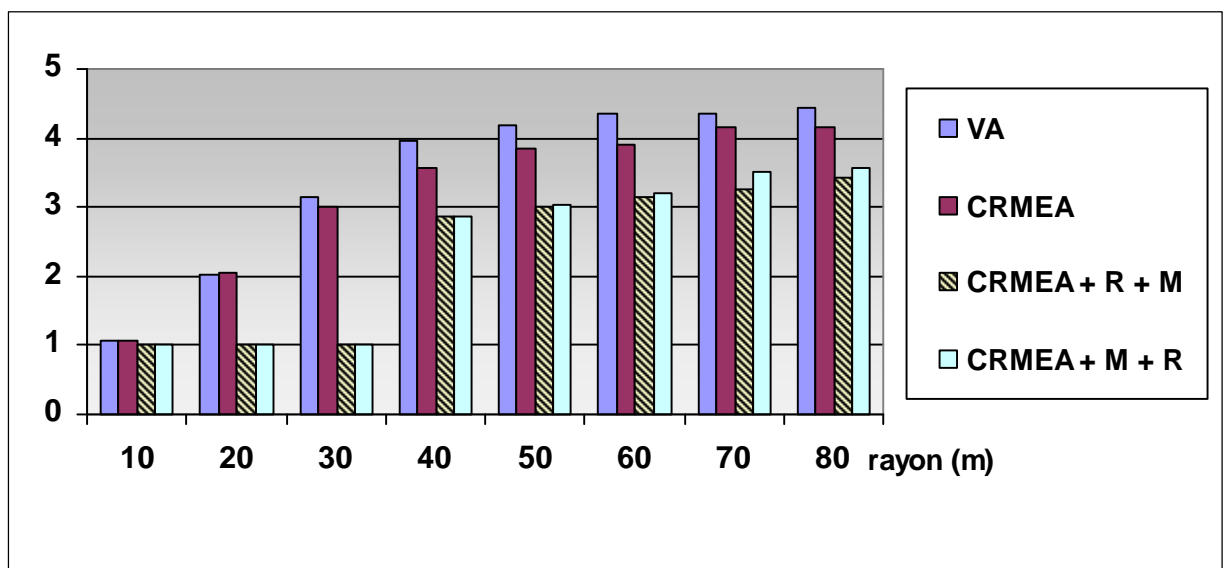


Figure 27 : Degré d'exclusion mutuelle en fonction du rayon d'évènement

#### IV.5. Degré maximal de couverture

La figure 28 représente l'impact de l'augmentation du rayon d'évènement sur le degré maximal de couverture pour les quatre stratégies : VA, CRMEA, CRMEA + R + M, et CRMEA + M + R.

On remarque que le degré maximal de couverture augmente en augmentant la taille de l'évènement.

Les résultats de ce facteur peuvent être justifiés de la même manière que le degré d'exclusion mutuelle.

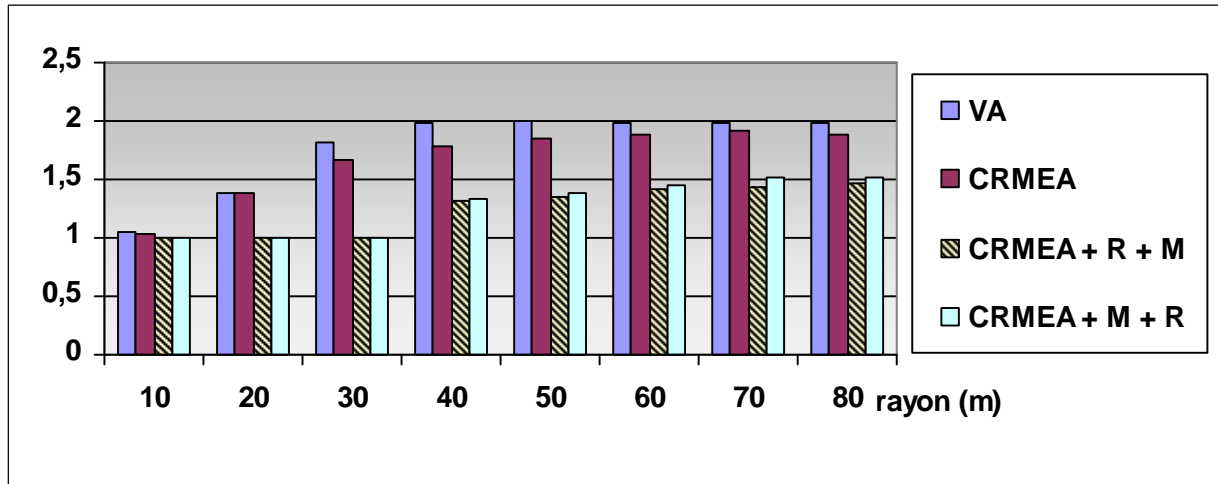


Figure 28 : Degré maximal de couverture en fonction du rayon d'évènement

#### IV.6. Délai de mobilité

La figure 29 représente l'impact de l'augmentation du rayon d'évènement sur le délai de mobilité (due à la mobilité) en supposant la vitesse de déplacement 1m/s pour les stratégies : CRMEA + R + M, et CRMEA + M + R.

- Pour les rayons moins de 30m, on remarque que les deux courbes sont superposées, cela peut être justifié par le fait que la mobilité déplace un seul actionneur pour couvrir la région d'évènement.

- Pour les rayons > 30m, on remarque que le délai de mobilité augmente de façon proportionnelle au rayon de l'évènement mais la stratégie CRMEA + R + M est meilleure que CRMEA + M + R.

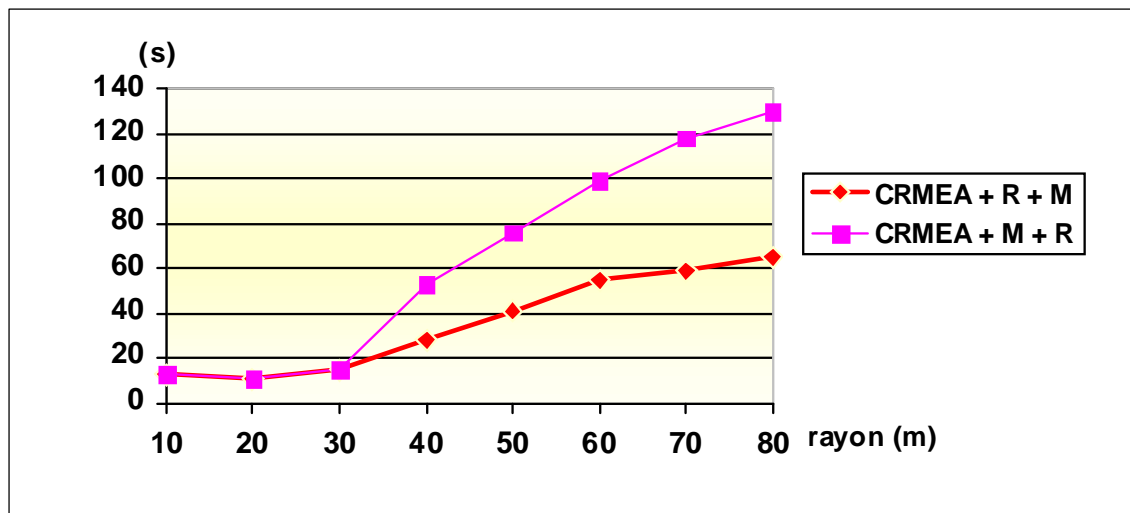


Figure 29 : Délai de mobilité en fonction du rayon d'évènement

#### IV.7. Energie de mobilité

La figure 30 représente l'impact de l'augmentation du rayon d'évènement sur l'énergie de mobilité (donnée par la formule  $\frac{1}{2} * m * V^2$ ) en supposant la vitesse de déplacement 1m/s pour les stratégies : CRMEA + R + M, et CRMEA + M + R.

- Pour les rayons moins de 30m, on remarque que les deux courbes sont superposées, cela peut être justifié par le fait que la mobilité déplace un seul actionneur pour couvrir la région d'évènement.

- Pour les rayons  $> 30m$ , on remarque que l'énergie de mobilité augmente de façon proportionnelle au rayon de l'évènement mais la stratégie CRMEA + R + M est meilleure que CRMEA + M + R.

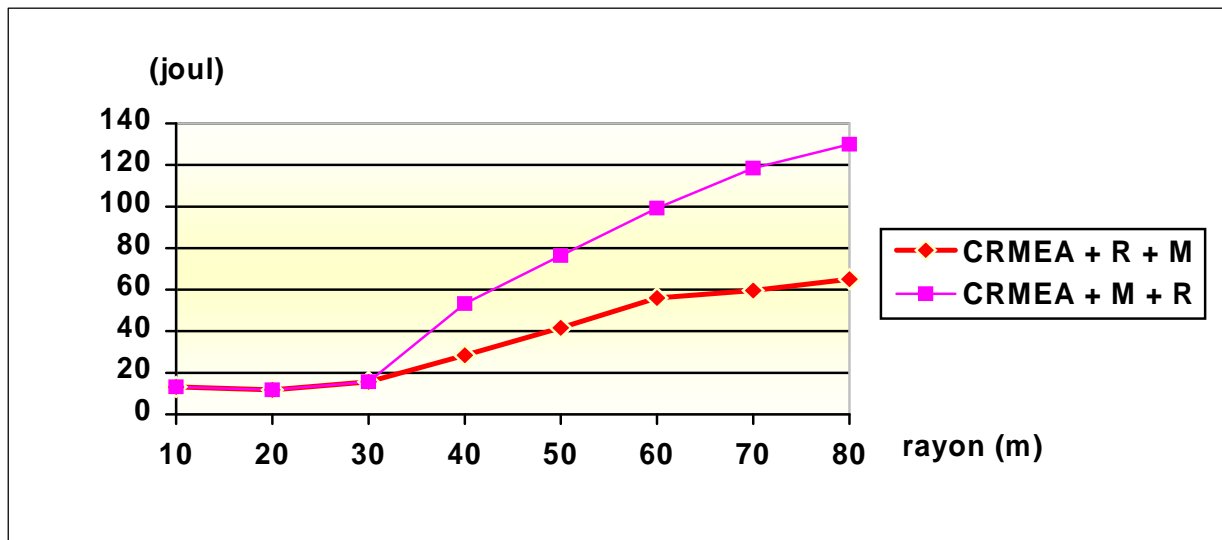


Figure 30 : Energie de mobilité en fonction du rayon d'évènement

#### IV.8. L'effet de la vitesse sur le délai de mobilité

La figure 31 représente l'impacte de l'augmentation du rayon d'évènement sur le temps d'action supplémentaire pour les stratégies : CRMEA + R + M, et CRMEA+ M + R.

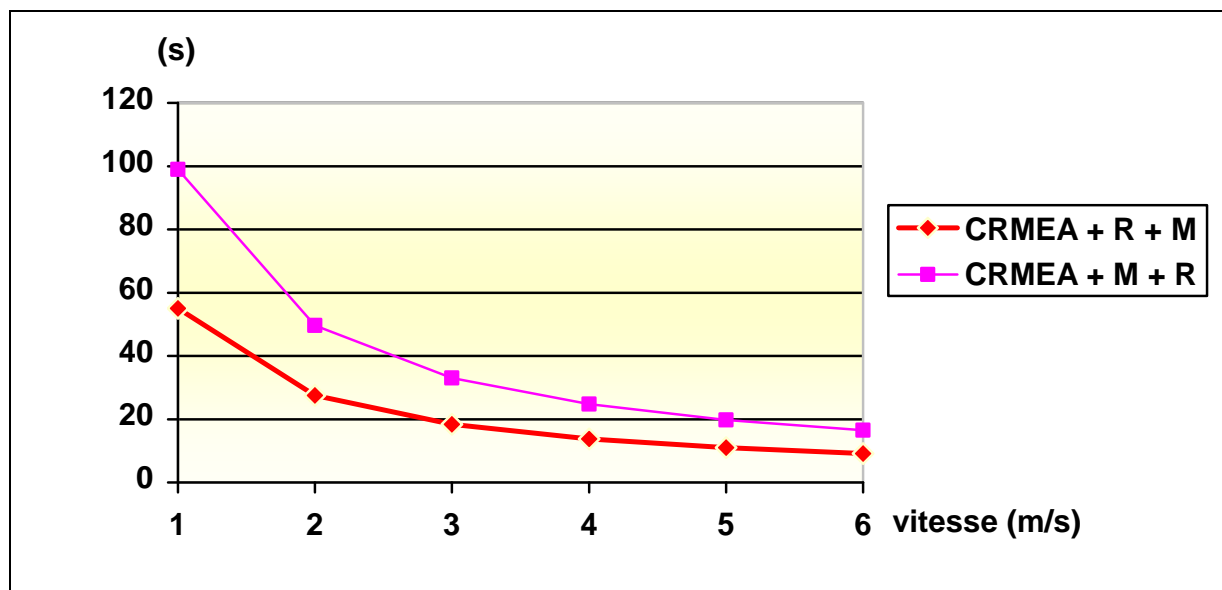


Figure 31 : Délai de mobilité en fonction de la vitesse des actionneurs (R=60m)

D'après la figure 31, on voit clairement qu'en augmentant la vitesse des actionneurs le délai nécessaire pour la mobilité sera de plus en plus court. De plus, le délai de mobilité

pour la stratégie CRMEA + M + R est plus important, car l'étape de réduction du rayon fait rapproché les actionneurs entre eux, d'où les distances entre les actionneurs sont plus petits.

#### IV.9. L'effet de la vitesse sur l'énergie de mobilité

La figure 32 représente l'impacte de l'augmentation de la vitesse des actionneurs sur l'énergie de mobilité pour les stratégies : CRMEA + R + M, et CRMEA + M + R.

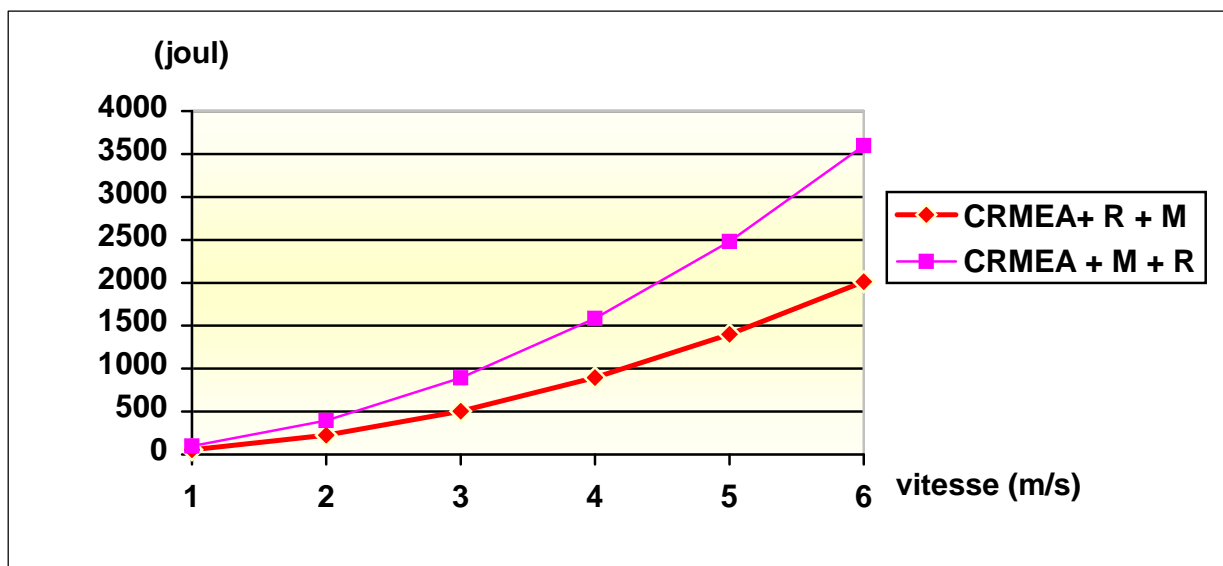


Figure 32 : Energie de mobilité en fonction de la vitesse des actionneurs (R=60m)

D'après la figure 32, l'énergie de mobilité est proportionnelle à la vitesse des actionneurs. De plus, on remarque que l'énergie dû à la mobilité est plus importante dans le cas de CRMEA + M + R. cela est justifié de la même manière que dans le paragraphe précédent.

## V. Conclusion

D'après les résultats de simulation, on constate que les différentes métriques augmentent en augmentant le rayon de l'évènement, cela est due au nombre d'actionneurs de couverture qui doit augmenter pour couvrir la région d'évènement. Quand le rayon de l'évènement est inférieur ou égale au rayon d'action,  $CRMEA + R + M$  et  $CRMEA + M + R$  donnent les mêmes résultats car dans ces cas un seul actionneur suffit pour couvrir la région d'évènement (mouvement d'un actionneur et ajustement de son rayon au rayon de la région d'évènement). Nous constatons que l'amélioration offerte par  $CRMEA + M$  est généralement meilleure que celle de  $CRMEA + R$ , et que  $CRMEA + R + M$  est meilleure que  $CRMEA + M + R$ . Finalement, on note que  $CRMEA + M + R$  requière un temps d'exécution plus important que celui de  $CRMEA + R + M$ .

# Conclusion générale

Les réseaux de capteurs et d'actionneurs constituent une très grande importance dans les présentes et futures applications grâce à leur faible coût, facilité de déploiement, et surtout les bénéfices qui peuvent apporter et les catastrophes qui peuvent être prévues et donc évitées. Dans ce travail, nous avons proposé une solution au problème d'exclusion mutuelle dans les réseaux WSANs. Nous avons constaté que la définition classique de l'exclusion mutuelle ne peut pas être appliquée sur les réseaux WSANs car les chevauchements entre les régions d'action sont inévitables.

Pour résoudre ce problème, nous avons proposé un algorithme d'exclusion mutuelle basé sur une approche centralisée (CRMEA), car cette dernière offre une couverture plus optimale que l'approche distribuée. Notre algorithme consiste à réduire les ressources supplémentaires dépensées sur la région d'évènement au lieu de minimiser le nombre d'actionneurs ou la surface chevauchée comme dans [1]. Nous avons aussi proposé des extensions supportant des rayons d'action dynamiques et des actionneurs mobiles. L'objectif de ces extensions est d'optimiser plus les ressources nécessaires pour couvrir une région d'évènement.

Pour valider notre solution, nous avons comparé par simulation CRMEA et ses extensions avec l'algorithme de Vedantham. Les résultats de simulation, montrent que nos stratégies sont meilleures en termes de coût d'action, énergie...etc. Cependant, les méthodes basées mobilité ont besoin d'un délai supplémentaire pour exécuter les actions. Une des perspectives de ce travail est d'étudier l'exclusion mutuelle dans les WSANs en considérant une approche distribuée.

## Bibliographie

- [1] Ramanuja Vedantham, Zhenyun Zhuang, and Raghupathy Sivakumar. “*Mutual exclusion in wireless sensor and actor networks*”. In 3rd Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON 2006), 2006.
- [2] Ian F. Akyildi and Ismail H. Kasimoglu. “*Wireless sensor and actor networks: research challenges*” Ad Hoc Networks Journal, 2(4):351-367, 2004.
- [3] Keiji Ozaki, Naohiro Hayashibara, and Makoto Takizawa “*Coordination Protocols of Multiple Actuator Nodes in a Multi-Actuator/Multi-Sensor Model*”, Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops (2007) - Volume 01 pp: 62-67.
- [4] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci “*Wireless sensor networks* ”, Computer Networks, Volume 38, Issue 4, 15 March 2002, Pages 393-422.
- [5] L. Lamport. “*Time, clocks and the ordering events in a distributed system*” Communication of the ACM, 21(7): 558-565, July 1978.
- [6] G. Ricart and A.K. Agrawala. “*An optimal algorithm for mutual exclusion in computer network*”, Communication of the ACM, 24(1): 9-17, January 1981.
- [7] M. Maekawa. “*A  $\sqrt{n}$  algorithm for mutual exclusion decentralized systems*”, ACM Transactions on Computer Systems, 3(2): 145-159, May 1985.
- [8] I. Suzuki and T. Kasami. “*A distributed mutual exclusion algorithm*”, Transactions on Computer Systems, 3(4): 344-349, November 1985.
- [9] K. Raymond. “*A tree Based algorithm for distributed mutual exclusion*”. ACM Transactions on Computer Systems, 7(1): 61-77, February 1989.
- [10] B. R. Badrinath, A. Acharya, and T. Imielinski, “*Structuring Distributed Algorithms for Mobile Hosts*”, In the 14<sup>th</sup> International Conference on Distributed Computing Systems, June 1994, pp 21-28.
- [11] Mukesh Singhal, D. Manivannan. “*A Distributed Mutual Exclusion Algorithm for Mobile Computing Environments*”, Proc of the IASTED Intel Conf on Intelligent Information, pp 557-561, 1997

- [12] D. Johnson, “*Approximation Algorithms for Combinatorial Problems*”, in *Journal of Computer and System Sciences*, 1974.
- [13] Uriel Feige, “*A Threshold of  $\ln n$  for Approximating Set Cover*,” in 28<sup>th</sup> ACM Symposium on Theory of Computing, Philadelphia, USA, May 1996, pp. 314–318.
- [14] R. D. Carr, S. Doddi, G. Konjevod, and M. V. Marathe, “*On the Red-Blue Set Cover Problem*” in Symposium on Discrete Algorithms, 2000, pp. 345–353.
- [15] Tommaso Melodia, Dario Pompili, Vehbi C. Gungor “*Communication and Coordination in Wireless Sensor and Actor Networks*”, *Ad Hoc Networks*, Volume 2, Issue 4, October 2004, Pages 351-367.
- [16] Kemal Akkaya, Mohamed Younis, “*COLA: A Coverage and Latency aware Actor Placement for Wireless Sensor and Actor Networks*”, in the Proceedings of IEEE Vehicular Technology Conference, 2006, Montreal, Que.
- [17] Brian McLaughlan and Kemal Akkaya, “*Coverage-based Clustering of Wireless Sensor and Actor Networks*”, *Journal of Parallel and Distributed Computing* Volume 69, Issue 6, June 2009, Pages 573-587
- [18] N. Bulusu, J. Heidemann, and D. Estrin, “*GPS-Less Low Cost Outdoor Localization for Very Small Devices*” in *IEEE Personal Communications, Special Issue on Smart Spaces and Environments*, Oct. 2000, pp. 28–34.
- [19] Arjan Durrezi, Vamsi Paruchuri, and Leonard Barolli. “*Delay-energy aware routing protocol for sensor and actor networks*”. In 11th International Conference on Parallel and Distributed Systems (ICPADS'05), 2005.
- [20] “Crossbow MICA2 Mote Specifications.” <http://www.xbow.com>.
- [21] “TinyOS community forum: An open-source OS for the networked sensor regime” <http://www.tinyos.net>
- [22] Tommaso Melodia, Dario Pompili, Ian F. Akyildiz “*A Communication Architecture for Mobile Wireless Sensor and Actor Networks*” sensor ad-hoc communication and networks (SECOM'06) sep 2006, Atlanta GA, pp.109-118.
- [23] B. Tavli, W. Heinzelman, “*TRACE: Time reservation using adaptive control for energy efficiency*”, *IEEE Journal on Selected Areas of Communication* 21 (10) (2003) 1506–1515.

- [24] M. Caccamo, L.Y. Zhang, L. Sha, G. Buttazzo, “*An implicit prioritized access protocol for wireless sensor networks*”, in: Proc. IEEE Real-Time Systems Symp, December 2002, pp. 39–48.
- [25] M. Conti, S. Giordano, G. Maselli, G. Turi, “*Cross-layering in mobile ad-hoc network design*”, IEEE Computer, Special Issue on AdHoc Networks 37 (2) (2004) 48–51.
- [26] A.J. Goldsmith, S. Wicker, “*Design challenges for energy constrained ad-hoc wireless networks*”, IEEE Wireless Communications 9 (4) (2002) 8–27.
- [27] I. Chlamtac, M. Conti, J.N. Liu, “*Mobile ad-hoc networking : imperatives and challenges*”, Ad Hoc Networks 1(1) (2003) 13–64.
- [28] Maxim Batalin, Gaurav Sukhatme, “*The Analysis of an Efficient Algorithm for Robot Coverage and Exploration based on Sensor Network Deployment*“, in the Proceedings of IEEE International Conference on Robotics and Automation, Barcelona, Spain, Apr 2005.
- [29] <http://robotics.jpl.nasa.gov/tasks/nrover/>
- [30] Maxim Batalin and Gaurav S. Sukhatme, “*Coverage, Exploration and Deployment by a Mobile Robot and Communication Network*”, in the Proceedings of the International Workshop on Information Processing in Sensor Networks, Palo Alto, CA, Apr. 2003.
- [31] M. Conti, S. Giordano, G. Maselli, G. Turi, “*Cross-layering in mobile ad-hoc network design*”, IEEE Computer, Special Issue on AdHoc Networks 37 (2) (2004) 48–51.
- [32] H.S. Kim, T.F. Abdelzaher, W.H. Kwon, “*Minimumenergy asynchronous dissemination to mobile sinks in wireless sensor networks*”, in: Proc. of the First ACM Int. Conf. on Embedded Networked Sensor Systems (ACM Sensys\_03), November 2003, pp. 193–204.
- [33] T. He, J. Stankovic, C. Lu, T. Abdelzaher, “*SPEED: A realtime routing protocol for sensor networks*”, in: Proc. IEEE Int. Conf. on Distributed Computing Systems (ICDCS), Rhode Island, USA, May 2003, pp. 46–55.
- [34] R. Baldoni, A. Virgillito, and R. Petrassi. “*A distributed mutual exclusion algorithm for mobile ad-hoc networks*”. In Proc of the 7th IEEE Symposium on Computer and Communications (ISCC 2002), 2002.

[35] M. Benchaïba , M. Haddad et M. Ahmed-Nacer, “*Une approche de solution au problème d’exclusion mutuelle dans les réseaux mobiles ad hoc*” , rapport de recherche, USTHB, Février 2006.

[36] Abdelouahid Derhab and Nadjib Badache. “*A distributed mutual exclusion algorithm over multi-routing protocol for mobile ad hoc networks*”. International Journal of Parallel, Emergent and Distributed Systems (IJPEDS), 23(3):197 { 218, 2008.

[37] J. Walter, J. Welch, and N. Vaidya, “*A k-mutual exclusion algorithm for ad hoc mobile networks*”, Dial M for Mobility workshop, Dallas, TX, USA, 1998, October.