

## Chapitre 4

# Couverture de cibles : formulation et approximation

### 4.1 Introduction

Lorsqu'un grand nombre de capteurs, avec de faibles ressources physiques, sont déployés aléatoirement dans une zone pour surveiller des points cibles, la gestion de la durée de vie du réseau est nécessaire dès lors que le remplacement des batteries épuisées n'est pas faisable. Le mécanisme le plus souvent étudié pour maintenir la couverture tout en maximisant la durée de vie du réseau, consiste en un ensemble de protocoles de planification du fonctionnement des capteurs, de façon à balancer leur activité entre deux états : actif et inactif. En étant actif, les modules de détection et communication du capteur sont opérationnels, et le capteur peut accomplir ses fonctions de capture et de communication. A l'opposé, un capteur à l'état inactif fonctionne avec un très bas niveau d'énergie.

Un premier niveau de gestion de l'énergie est le design des éléments matériels composant le noeud du capteur. Des efforts considérables ont été apportés ces dernières années, allant dans le sens de l'optimisation de la consommation en énergie des capteurs, tout en maintenant un fonctionnement optimal de ces composants électroniques. Cependant, une autre technique conventionnellement adoptée dans les travaux relatifs à la question, est la réduction du cycle de fonctionnement. Ce dernier est défini par le ratio entre la durée de fonctionnement à l'état actif, et la durée pendant laquelle le capteur est à l'état inactif.

Maximiser la durée de vie d'un réseau de capteurs (avec un grand nombre de capteurs), tout en satisfaisant une certaine qualité de couverture, est un problème d'optimisation, qui la plupart du temps ne peut pas être résolu de manière optimale et sur un délai de temps raisonnable. Dans ce chapitre, nous abordons ce problème (en section 2) en utilisant l'approche des ensembles couvrants, présentée en section 3. Nous proposons une technique gloutonne pour générer des ensembles couvrants, sous forme d'algorithme avec un rapport d'approximation inférieur à  $\log(m)$ , avec  $m$  le nombre de cibles (section 4). Ensuite, nous comparons, par simulation, notre

proposition avec des travaux similaires évoqués dans le chapitre précédent [78, 101]. Enfin, nous présentons une interprétation des résultats de simulation, avant de conclure sur les objectifs de ce chapitre.

## 4.2 Formulation du problème

### 4.2.1 Hypothèses

Nous considérons un scénario où les capteurs sont dispersés aléatoirement à proximité de cibles localisées à des positions connues dans un champ, comme montré dans la figure 4.1. Le déploiement satisfait la contrainte selon laquelle chaque cible est couverte par au moins un capteur après le déploiement. Chaque capteur possède un rayon de détection prédéfini  $r$ , une source d'alimentation en énergie et est censé être équipé du module requis pour localiser sa position géographique. Nous considérons un modèle de couverture circulaire, c'est-à-dire que la mesure de couverture entre un capteur et une cible est égale à 1 si la distance entre le capteur et la cible est inférieure à  $r$ , et 0 sinon. Un capteur couvre une cible si la distance euclidienne entre eux est inférieure au rayon de détection. Nous supposons également l'existence d'un contrôleur dans une station de base qui dispose de la position des capteurs et des cibles et centralise les protocoles du réseau. Chaque capteur est également équipé d'un module de contrôle d'énergie afin qu'il puisse alterner entre les états actif et inactif. Lorsqu'il est actif, le capteur consomme de l'énergie pour détecter et traiter les données physiques recueillies à partir de cibles surveillées et communiquer éventuellement. L'état inactif permet au capteur de conserver l'énergie sans pouvoir détecter ou communiquer des données, à l'exception de la réception de messages de synchronisation envoyés par le coordonnateur. Le problème de connectivité n'est pas pris en compte. Par conséquent, il est considéré que les capteurs sont capables de transmettre des données collectées en utilisant une communication directe vers la station de base.

### 4.2.2 Définitions

#### Problème de couverture cible

Considérant les hypothèses énoncées ci-dessus, le problème de couverture de cibles consiste à planifier les activités des capteurs de manière à ce que toutes les cibles soient constamment observées et que la durée de vie soit rallongée au maximum possible. Étant donné que le déploiement garantit que chaque cible est couverte par au moins un capteur et que les cibles sont couvertes de manière redondante, les capteurs peuvent être divisés en différents groupes, chacun satisfaisant les contraintes de couverture. Les groupes sont activés un à un, et seul le groupe activé contient les capteurs actifs dans le champ, tandis que les capteurs dans les autres groupes sont inactifs. Dans ce qui suit, nous proposons de résoudre ce problème en l'abordant sous la formulation du problème des ensembles couvrants.

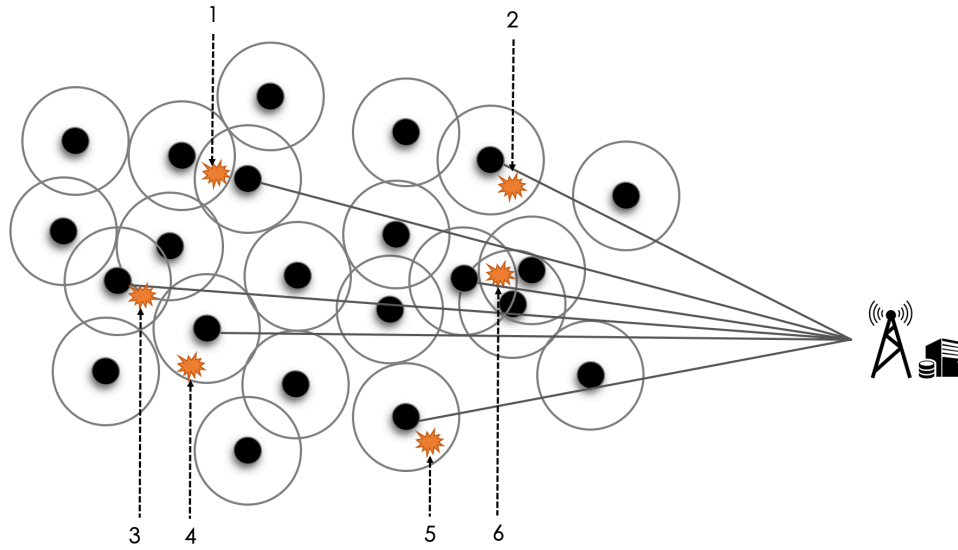


FIGURE 4.1 – Illustration du réseau de capteurs sans fil.

### Problème des ensembles couvrants

Étant donné un univers représenté par l'ensemble  $U$ , et une collection  $S$  de sous-ensembles de  $U$ , un ensemble couvrant est définie comme une sous-collection  $C \in S$  d'ensembles dont l'union est égale à  $U$ . Après le déploiement, chaque capteur (correspondant à un élément de la collection  $S$ ), couvre un ensemble de cibles (représenté par les éléments de  $U$ ). L'objectif principal du problème des ensembles couvrants dans la couverture de cibles revient donc à trouver un sous-ensemble de capteurs de cardinalité minimale couvrant toutes les cibles. La figure 4.1 en donne une illustration. Six cibles sont dans la topologie schématiquement représentée. Chaque cible est sous la couverture d'au moins un capteur après le déploiement. Certaines cibles comme la cible 1 sont couvertes par 2 capteurs, alors que la cible 6 est couverte par 3 capteurs.

Afin de maximiser la durée de vie du réseau, une proposition intuitive serait de maximiser le nombre d'ensembles couvrants en tentant de minimiser le nombre de capteurs inclus dans chaque ensemble couvrant. Ce problème a été abordé dans [20, 39], et fait partie des problèmes les plus anciens et les plus étudiés dans la classe des problèmes NP-difficiles.

### Contraintes du problème

Nous ajoutons des contraintes au problème de trouver un nombre maximum d'ensembles couvrants. Les ensembles peuvent être disjoints ou non-disjoints. Dans le premier cas, un capteur ne peut être inclus dans plus d'un ensemble, alors que pour le second le capteur peut se retrouver dans plusieurs ensembles et pourra être activé plusieurs fois. Un capteur peut être inclus dans plusieurs ensembles, à condition qu'il n'ait pas épuisé sa source d'énergie.

## 4.3 Mécanisme de sélection des capteurs

### 4.3.1 Maximum théorique

Le caractère aléatoire du déploiement associé à l'effet de redondance dans la couverture peut entraîner une disproportion sur le nombre de capteurs couvrant chaque cible. Nous désignons par  $S_j$  le sous-ensemble des capteurs couvrant la cible  $z_j$  et  $z_k$  le nombre de cibles couvertes par le capteur  $s_k$ . Après le déploiement, certaines cibles peuvent se retrouver plus faiblement couvertes que d'autres. On note  $z_{min}$  la cible couverte par le plus petit nombre de capteurs après le déploiement.  $z_{min}$  est définie comme étant la cible critique. Les capteurs qui le couvrent sont appelés capteurs critiques. Le nombre de capteurs qui couvrent la cible critique place une limite supérieure sur le nombre d'ensembles couvrants calculables, à savoir le maximum théorique. La relation entre le nombre d'ensembles calculables et le maximum théorique est décrite dans l'équation 4.1.

$$|C(I)| = d \times z_{min} \quad (4.1)$$

Où  $C(I)$  dénote la collection d'ensembles calculée par un algorithme d'approximation pour une instance  $I$  du problème, et  $d$  le nombre maximum d'ensembles dont un capteur peut être membre. Pour calculer le nombre d'ensembles couvrants le plus proche du maximum théorique, le nombre de capteurs dans un ensemble doit être aussi minimal que possible. Ceci établit clairement que la maximisation des ensembles est conduite par la minimisation du nombre de capteurs dans chaque ensembles couvrants. Par conséquent, une stratégie efficace de sélection des capteurs doit être adoptée. Cette dernière doit éviter d'inclure plus d'un capteur couvrant le/les cible(s) critique(s) dans un ensemble. Une sélection précise peut être effectuée en définissant une fonction de profit qui représente la contribution de chaque capteur par rapport à l'ensemble des cibles.

### 4.3.2 Évaluation du choix des capteurs

Chaque capteur a une contribution, qui peut être considérée comme sa valeur de rentabilité en relation avec d'autres capteurs et les cibles dans le réseau. Les zones de détection des capteurs se chevauchent et, par conséquent, les cibles peuvent être couvertes de manière très dense. Dans la conception de notre algorithme, nous donnons la priorité aux capteurs qui couvrent les cibles les moins couvertes et couvrant un nombre maximal de cibles non encore couvertes. Par exemple, lors de la construction d'un ensemble, une cible peut être couverte par un seul capteur. Ce capteur sera membre de cet ensemble, peu importe le nombre de cibles qu'il couvre. Le processus de sélection des capteurs dans ce premier cas est trivial. Dans le second cas, ce qui signifie que les cibles sont simultanément couvertes par plusieurs capteurs, le capteur le plus rentable est sélectionné. Le rapport coût-efficacité d'un capteur  $s_k$  peut être mesuré par rapport au nombre de cibles couvertes et la rentabilité d'autres capteurs qui chevauchent la zone de détection de  $s_k$ . Nous introduisons d'abord la fonction de relation de couverture (CRF), qui décrit l'état de

couverture d'un capteur. Le CRF entre  $s_k$  et  $z_j$  est égal au rapport entre la valeur de détection de  $s_k$  et  $z_j$  et la somme des valeurs de détection de tous les capteurs couvrant  $z_j$ . Si ces derniers peuvent surveiller  $z_j$  avec une valeur relativement grande, cela implique que  $z_j$  est densément couvert, ce qui entraîne un CRF inférieur pour  $s_k$  contrairement au cas où  $z_j$  serait faiblement couverte. Nous définissons le CRF dans l'équation 4.2.

$$w_j^k = \frac{p_j^k}{\sum_{s_i \in S_0} p_j^i} \quad (4.2)$$

Où  $p_k^j$  est une fonction binaire égale à 1 si la cible  $z_j$  est couverte par  $s_k$ , et 0 sinon,  $S_0$  représente l'ensemble complet des capteurs. En considérant  $Z_0$  comme l'ensemble complet des cibles, la rentabilité de  $s_k$  par rapport à  $Z_0$  est égale à la somme normalisée des CRF entre  $s_k$  et toutes les cibles appartenant à  $Z_0$ .

$$w_{Z_0}^k = \sum_{z_j \in Z_0} \frac{p_j^k}{m \sum_{s_i \in S_0} p_j^i} \quad (4.3)$$

Lors de la construction des ensembles couvrants, les capteurs peuvent être sélectionnés de manière itérative, en appliquant l'équation 4.3 à chaque étape pour sélectionner le capteur au profit maximal.

## 4.4 Technique d'approximation

Dans cette section, nous proposons d'abord un algorithme glouton qui tente de sélectionner un nombre minimal de capteurs pour former un ensemble couvrant. Ensuite, nous présentons l'algorithme glouton de génération d'ensembles couvrants successifs, dont le rôle est d'itérer le processus de construction des ensembles couvrants avec les capteurs disponibles.

### 4.4.1 Ensemble couvrant minimal

Nous commençons par introduire un algorithme glouton qui repose essentiellement sur l'application de l'équation 4.3 pour former un ensemble couvrant. Le pseudo-code de l'algorithme est donné dans l'algorithme 1. L'algorithme est composé d'une itération pendant laquelle l'ensemble des cibles non couvertes est d'abord initialisé à l'ensemble complet de cibles, avant de commencer à sélectionner successivement le capteur ayant la plus grande valeur de rapport coût-efficacité selon l'équation (3). Une fois qu'un capteur a été sélectionné, les cibles couvertes par ce capteur sont retirées de la liste ( $Uncvd$ ) initialisé au départ et le capteur est ajouté à l'ensemble en cours de construction. Le processus est réitéré jusqu'à ce toutes cibles soient couvertes, i.e. ( $Uncvd = \phi$ ).

**Algorithme 1** : Algorithme glouton de calcul d'un ensemble couvrant minimal

---

```

input   :  $S_0 = (s_1, \dots, s_n, Z_0 = z, \dots, z_m)$ 
output :  $C$ 
 $Uncvd = Z_0;$  1
 $S_{cur} = S_0;$  2
 $C = \phi;$  3
while  $Uncvd \neq \phi$  do 4
     $\forall s_k \in S_0;$  5
    Évaluer la fonction  $w_{Z_0}^k;$  6
    Sélectionner le capteur au profit maximal ( $s_{max}$ ); 7
     $C = C \cup s_k;$  8
     $Uncvd = Uncvd - Z_{max};$  9
end 10
return  $C;$  11

```

---

**4.4.2 Rapport d'approximation**

*Proposition.* Le rapport d'approximation de *Ensemble couvrant minimal* est de l'ordre de  $\log(m)$ .

*Preuve.* Soit  $m_t$  le nombre de cibles non couvertes après  $t$  itérations de l'algorithme et  $k$  la cardinalité d'un ensemble couvrant calculé par la solution optimale  $Opt(I)$ , pour une instance  $I$  du problème. Alors  $Opt(I)$  effectue  $k$  itérations pour calculer l'ensemble couvrant. A la  $t_{ime}$  itération, étant donné que chaque cible restante est couverte par au moins un des  $k$  capteurs, l'un des capteurs disponibles dans  $Opt(I)$  couvre au moins  $m_t/k$  cibles. Cela implique que :

$$m_{t+1} \leq \frac{m_t}{k} \quad (4.4)$$

$$m_t \leq m_0 \left(1 - \frac{1}{k}\right)^t, \forall t, 1 \leq t \leq k \quad (4.5)$$

L'équation 4.5 indique que le nombre de cibles restantes à la  $t_{ime}$  itération reste inférieur à la cardinalité de l'ensemble complet de cibles. De plus, il peut être prouvé (une simple représentation graphique 2D de ces deux fonctions suffit) que :

$$1 - X \leq \exp(-X) \quad (4.6)$$

Où  $(1 - X) = \exp(-X)$  si  $X = 0$  et  $(1 - X) \leq \exp(-X)$  pour toutes les valeurs de  $X \neq 0$ .

Si  $X = \frac{1}{k}$ , alors nous pouvons combiner 4.5 et 4.6 pour obtenir :

$$\left(1 - \frac{1}{k}\right)^t < \left(\exp\left(-\frac{1}{k}\right)\right)^t \quad (4.7)$$

$$\Rightarrow m_t \leq m \left(1 - \frac{1}{k}\right)^t < m \left(\exp\left(-\frac{1}{k}\right)\right)^t \quad (4.8)$$

Quand  $t = k \ln(m)$

$$m_{k \ln(m)} < m(\exp(-\ln(m))) = 1 \quad (4.9)$$

Par conséquent, chaque cible est couverte après au plus  $k \ln(m)$  itérations de notre algorithme glouton. Le nombre de capteurs inclus dans un ensemble couvrant calculé par notre algorithme glouton est  $\ln(m)$  fois celui calculé par  $Opt(I)$ .

#### 4.4.3 Génération d'ensembles couvrants

L'algorithme de génération d'ensembles couvrants est montré dans 2. Il procède de manière itérative, en vérifiant à chaque tour la disponibilité des capteurs à pouvoir former un nouvel ensemble couvrant. Si c'est le cas, l'Algorithme 1 est exécuté pour former un nouvel ensemble couvrant. A la suite de celui-ci, les quantités d'énergie des capteurs, sont mises à jour avant d'entamer la construction d'un nouvel ensemble.

---

**Algorithme 2** : Algorithme itératif de calcul d'ensembles couvrants successifs.

---

<b>input</b> : $S_0 = (s_1, \dots, s_n, Z_0 = z_1, \dots, z_m)$	
<b>output</b> : $C$	
$c = 0;$	1
$S_{cur} = S_0;$	2
$C = \phi;$	3
$C_c = \phi;$	4
<b>while</b> $S_{cur} \neq \phi$ <b>do</b>	5
Exécuter l'algorithme 1 sur $S_{sur}$ , $Z_0$ et $C_c$ ;	6
Mettre à jour le niveau d'énergie des capteurs sélectionnés;	7
$C = C \cup C_c;$	8
$c = c + 1;$	9
<b>end</b>	10
return $C$ ;	11

---

#### 4.4.4 Analyse de la complexité

Considérons que nous ayons  $n$  capteurs déployés aléatoirement dans une zone d'intérêt pour surveiller  $m$  cibles. Lors de la construction d'un ensemble couvrant, les capteurs sont sélectionnés de façon itérative jusqu'à ce que toutes les cibles soient couvertes. Lorsque la construction de l'ensemble est terminée, toutes les durées de vie des capteurs sélectionnés sont mises à jour. Si la durée de vie restante d'un capteur lui permet de faire partie d'un autre ensemble, ce capteur peut être sélectionné ultérieurement. Sachant que chaque capteur dépense uniformément  $b$  unités d'énergie lorsque son ensemble est activé, un capteur peut être sélectionné au plus  $d$  fois par l'algorithme, avec  $d = \frac{e^k}{b}$ ,  $\forall s_i \in S_0$ , où  $e_k$  représente la durée de vie initiale de la batterie. L'exécution au pire des cas aurait inclus l'ensemble des  $n$  capteurs, chacun d'entre eux étant

utilisé  $d$  fois avec  $m$  cibles. Le temps d'exécution de l'algorithme est équivalent à la sommation suivante :

$$T(n) = dnm + d(n-1)(m-1) + \dots + d(n(n-1)(m - (n-1) \bmod m)) \quad (4.10)$$

$$T(n) = d \sum_{k=0}^{n-1} (n-k)(m - k \bmod m) \quad (4.11)$$

Nous pouvons en déduire que la complexité au pire des cas de l'algorithme de génération d'ensembles couvrants est égale à  $\mathcal{O}(dn^2m)$ .

## 4.5 Tests et simulations

Pour évaluer la performance de notre algorithme, nous simulons un réseau avec des nœuds capteurs statiques et des cibles localisées aléatoirement dans une zone virtuelle de 100x100m. Un prototype d'implémentation de l'Algorithme 2 est développé sur MATLAB 7.0. Pour la génération de la topologie, les cibles sont réparties en premier dans la zone en générant un couple de coordonnées aléatoires pour chaque cible, suivies des capteurs de sorte que chaque cible soit couverte par au moins un capteur.

### 4.5.1 Ensembles disjoints

Dans la première expérience, nous faisons varier le nombre de capteurs de 30 à 100, fixant le nombre de cibles à 10, et la portée de détection à 30 m. L'énergie initiale des capteurs est fixée à 1J et la durée de vie de chaque ensemble de couverture est égale à 0,5 unité de temps. Un modèle de consommation d'énergie linéaire est considéré pour simplification, c'est-à-dire si un capteur est actif pour  $e$  unités de temps, alors il consomme  $e$  unités d'énergie. Les algorithmes sont exécutés 10 fois avec différentes instances de positionnement des cibles et des capteurs, et les valeurs moyennes du nombre d'ensembles sont calculées et représentées sur la figure 4.2.

Nous commençons par comparer nos résultats avec le maximum théorique. Parallèlement à l'évolution du nombre d'ensembles disjoints lorsque le nombre de capteurs augmente, on peut remarquer la proximité des résultats produits par notre algorithme avec le maximum théorique. Ces résultats mettent en évidence la flexibilité de notre fonction de profit lors de la sélection des capteurs. Les cibles critiques et les capteurs critiques correspondants ont été bien gérés.

A côté de la proximité de notre algorithme avec le maximum théorique, nous comparons d'autre part la moyenne du nombre d'ensembles couvrants disjoints calculés par l'Algorithme 2, avec l'algorithme MC-MCh<sup>1</sup> de Slijepcevic et Potkonjak[78] et B{GoP} de Zorbas[101]. En considérant les ensembles couvrants disjoints, les résultats de simulation montrent à travers l'augmentation du nombre de capteurs, une évolution constante et une tendance de supériorité

---

1. Minimum Constraining heuristic



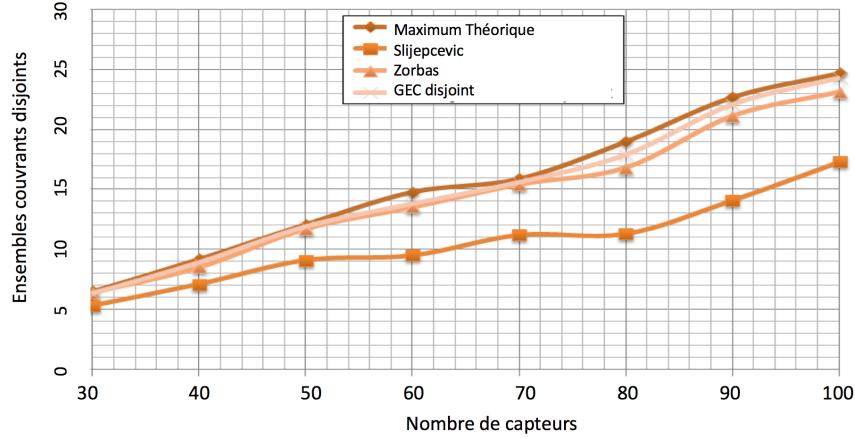


FIGURE 4.2 – Evolution du nombre d'ensembles disjoints avec le nombre de capteurs.

en termes d'ensembles couvrants lorsque nous comparons notre algorithme avec les deux autres. La distance entre les résultats de [78] et de [101] montre que la stratégie adoptée pour gérer les cibles critiques ne garantit pas toujours l'absence de cibles doublement couvertes, ou couvertes de manière redondante. Ainsi, chaque fois qu'un nouvel ensemble est construit, il y a eu au moins une cible couverte plus d'une fois. Cela devient plus compliqué lorsque cette cible est critique. Si une cible critique est couverte par  $k$  capteurs ( $k \geq 0$ ) dans un ensemble couvrant, alors  $k$  ensembles seront sacrifiés et il sera impossible d'atteindre le maximum théorique.

#### 4.5.2 Ensembles non-disjoints

Nous avons utilisé les mêmes paramètres que celles de l'expérience précédente et tracé les résultats dans la figure 4.3 après 10 exécutions successives des algorithmes évalués. On peut remarquer que le nombre d'ensembles non disjoints calculés par les algorithmes à partir d'un nombre fixe de capteurs est deux fois supérieur au nombre d'ensembles disjoints. Par exemple, avec 30 et 100 capteurs déployés, les cardinalités correspondantes sont respectivement 6 et 25 en considérant les ensembles disjoints, et 16 et 61 en considérant des ensembles non disjoints. Sachant qu'un capteur inclus dans des ensembles non disjoints est susceptible d'être activé  $d$  fois, ce capteur sera actif pour les unités de temps  $e_k^i = \frac{e_k}{d}$ , tandis qu'un capteur dans un ensemble disjoint sera activé une fois pour fonctionner pour  $e_k^i = e_k$  unités de temps. Avec un tel arrangement, nous pouvons nous attendre à une durée de vie plus longue dans le cas des ensembles couvrants non-disjoints.

#### 4.5.3 Durée de vie du réseau

Dans ce troisième scénario, nous étudions l'évolution de la durée de vie du réseau par rapport à l'augmentation du nombre de capteurs. La station de base exécute l'algorithme et pour activer un ensemble de capteurs sélectionnés pour un ensemble couvrant, et envoie un message de démarrage aux capteurs qui sont membres d'un ensemble. Dans le cas des ensembles disjoints, ces capteurs

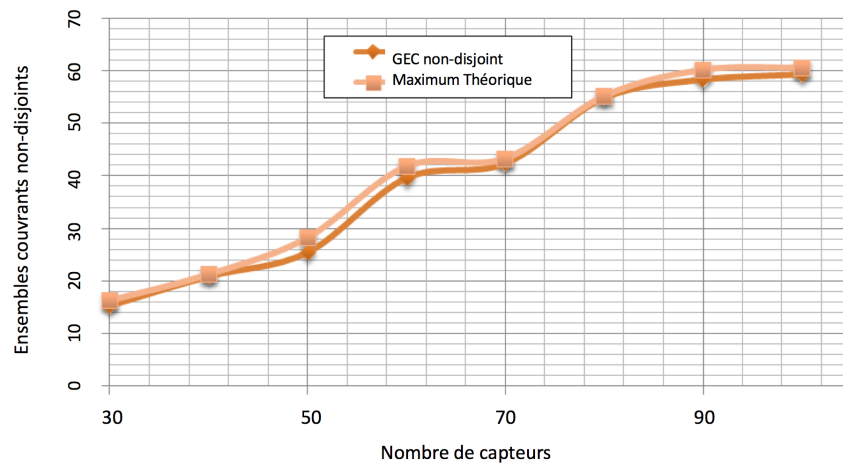


FIGURE 4.3 – Evolution du nombre d'ensembles non-disjoints avec le nombre de capteurs.

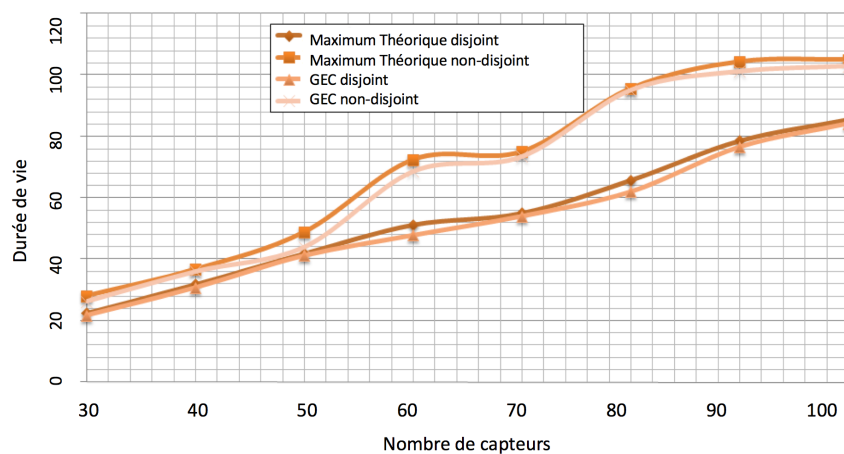


FIGURE 4.4 – Durée de vie du réseau en fonction du nombre de capteurs.

sont actifs jusqu'à ce que leur réserve d'énergie atteigne 0. Lorsque l'énergie initiale de tous les capteurs est la même, tous les ensembles disjoints partagent la même durée d'opération. Cette dernière est alors considérée comme linéairement proportionnel à l'énergie des capteurs. La durée de vie du réseau en utilisant des ensembles disjoints est égale à la durée de vie d'un ensemble multiplié par le nombre d'ensembles disjoints. La figure 4.4 montre que la durée de vie du réseau est fortement améliorée lorsque des ensembles non-disjoints sont produits. Cette remarque devient plus intéressante lorsque le nombre de capteurs augmente. Par exemple, avec 30 et 100 capteurs, la durée de vie est augmentée de 4 à 18. La contrainte de non-disjonction associée à la redondance de couverture des cibles peut améliorer la durée de vie du réseau, en raison de l'implication des capteurs dans plusieurs ensembles couvrants.

#### 4.5.4 Quantité de cibles et rayon de détection

Pour cette dernière expérience, nous faisons varier le nombre de cibles et le rayon de détection pour observer l'impact sur le nombre d'ensembles. La remarque générale est que la même tendance

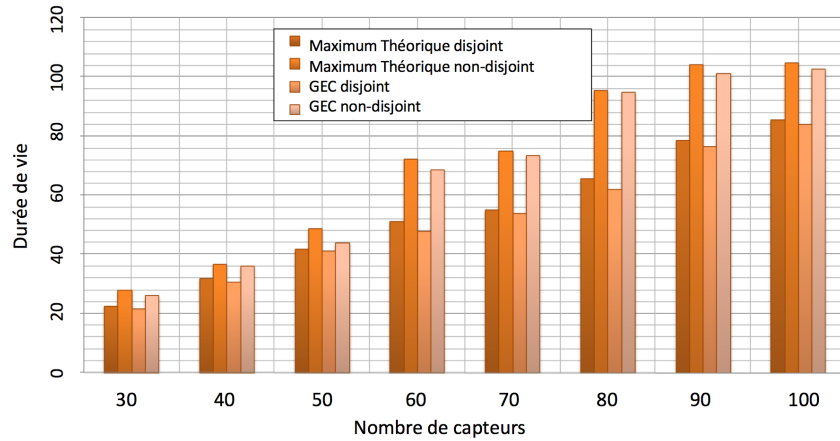


FIGURE 4.5 – Evolution du nombre d’ensembles en fonction du nombre de cibles et du rayon de détection des capteurs.

est conservée lorsqu’on compare le nombre d’ensembles disjoints calculés par les algorithmes (figure 4.5). Cependant, lorsque la portée de détection augmente, cela peut améliorer de manière constante le nombre d’ensembles puisque chaque capteur pourra couvrir plus de cibles, comparé à des rayons fixes. Cela induit à une plus forte redondance dans la couverture, qui peut conduire à une augmentation du nombre de capteurs qui couvrent les cibles critiques. Lorsque le nombre de cibles augmente pendant que le nombre de capteurs est fixe, la probabilité de la redondance de la couverture est susceptible d’être réduite et le nombre d’ensembles pourrait également diminuer.

#### 4.5.5 Quelques aspects d’efficacité

A travers ces résultats énoncés ci-haut, la flexibilité de la fonction de profit pour la sélection des capteurs lors de la construction des ensembles couvrants est mise en évidence. Étant donné que l’algorithme fonctionne à travers plusieurs itérations, il y a juste un paramètre de cette fonction qui est susceptible de changer : l’énergie restante des capteurs dont les zones de détection chevauchent la zone de détection du capteur évalué. De plus, cette flexibilité prend implicitement en compte les capteurs qui couvrent la cible critique. Il n’est pas nécessaire d’identifier d’abord le/les cible(s) critique(s) avant de choisir parmi le/les capteur(s) critique(s) celui qui couvre le nombre maximum de cibles non encore couvertes au cours de cette étape, comme dans [101]. Le cas idéal est présenté lorsqu’un capteur couvre un grand nombre de cibles non couvertes et qu’il est le seul capteur couvrant ces cibles. Dans ce cas, la valeur de la fonction de rentabilité sera égale à 1. Cela signifie que toutes les cibles couvertes par ce capteur sont critiques. Même s’il existe une cible critique dans le réseau, le capteur correspondant sera sélectionné, quel que soit le nombre de cibles non encore couvertes que ce capteur couvre. Le cas le plus difficile se présente lorsque plusieurs capteurs non sélectionnés couvrent le même nombre de cibles non couvertes et qu’il y a une forte redondance dans la couverture des cibles. Dans ce cas, presque tous les capteurs évalués ont les mêmes valeurs et le capteur ayant le plus petit indice est sélectionné. La sélection dans ce dernier cas de figure devient aléatoire.

## 4.6 Conclusion

Dans ce chapitre, nous avons abordé le problème de la couverture de cibles dans les réseaux de capteurs sans fil. Lorsque plusieurs capteurs alimentés par batteries sont dispersés dans une zone pour surveiller un ensemble fini de cibles, la gestion et le contrôle de l'énergie des capteurs est une nécessité pour obtenir un réseau à longue durée de vie. Un mécanisme bien connu consiste à planifier les activités des capteurs, de manière à ne garder actif que le minimum nécessaire pour satisfaire les besoins en couverture. Pour y parvenir, nous avons utilisé l'approche des ensembles couvrants et proposé un algorithme glouton qui sélectionne le nombre minimal de capteurs pour couvrir l'ensemble des cibles. Même si des solutions intéressantes ont été proposées au préalable, elles supposent généralement beaucoup de considérations qui rendent l'étude impraticable avec des réseaux de capteurs réalistes. Nous sommes motivés à explorer, dans les chapitres suivants, d'autres aspects de la question avec notamment un regard sur les modèles probabilistes.

## Chapitre 5

# Couverture de cibles sur un modèle probabiliste

### 5.1 Introduction

Bien que plusieurs solutions au problème de la couverture cible ont été proposées, la plupart est basée sur une hypothèse d'une couverture parfaite entre un capteur et une cible, et ne clarifient pas la façon dont chaque cible est couverte par les capteurs voisins [78, 15, 14, 101, 102]. Un autre point faible est l'hypothèse selon laquelle les modèles de couverture utilisés sont des modèles de couverture binaires, c'est-à-dire qu'une cible est couverte par un capteur si sa position est incluse à l'intérieur du disque délimitant la zone de couverture de ce dernier.

Notre travail dans ce chapitre est décrit comme suit : i) nous abordons le problème de la couverture cible en utilisant une approche des ensembles couvrants. Nous proposons un schéma de couverture qui tente de maximiser le nombre d'ensembles couvrants disjoints et non-disjoints, à partir de la population des capteurs aléatoirement déployés pour surveiller les cibles. L'algorithme proposé est un raffinement de nos travaux précédents dans [30], où les ensembles couvrants produits étaient disjoints. ii) Nous tentons d'apporter des considérations pratiques dans la couverture de cible associée au problème de la maximisation de la durée de vie, lors de l'organisation des capteurs en ensembles couvrants. La détection de chaque cible suit un modèle probabiliste, et sa mesure est une valeur réelle positive normalisée entre 0 et 1. Une fonction de profit probabiliste est proposée, qui permet de sélectionner les capteurs à inclure dans les ensembles couvrants.

Le reste du chapitre est organisé comme suit. En section 2, nous présentons le modèle de probabiliste utilisé comme modèle de couverture entre un capteur et une cible, et un modèle de détection collective lorsque plusieurs capteurs couvrent une même cible simultanément. En section 3, nous présentons une vue formelle du problème, avec une programmation linéaire d'entiers proposée en section 4, pour mettre en évidence les différentes variables se rapportant au problème. Nous exposons deux solutions approximatives en section 5 sous forme d'algorithmes présentés en pseudo-codes : la première est un algorithme de recherche local, alors que la deuxième se base sur

une approche gloutonne de génération d'ensembles couvrants avec application d'une fonction de profit. L'évaluation des performances des algorithmes proposés est présentée en section 6, avant de conclure sur les objectifs de ce chapitre.

## 5.2 Modèle de détection de cible

### 5.2.1 Détection individuelle

Considérons un capteur  $s_k$  et une cible  $z_j$ , émettant un signal uniforme et omnidirectionnel. La fonction de propagation  $P_f$ , d'un signal se propageant en espace libre, et provenant d'une source (représentant une cible) peut être représentée par l'équation 5.1 :

$$P_f \left( O \left( \frac{\theta}{d_{k,j}}, \alpha \right), \mathcal{H}(\eta) \right) = \frac{\theta}{d_{k,j}^\alpha} + \mathcal{H}(\eta) \quad (5.1)$$

$\theta$  représente la puissance du signal,  $\alpha$  modélise un coefficient d'atténuation du signal qui peut être caractéristique du type d'environnement, et  $d_{k,j}$  la distance entre  $s_k$  et  $z_j$ . La première partie de l'équation 5.1, comme représentée dans 5.2, décrit la fonction de variation du signal par rapport à la distance, le degré d'atténuation, et l'amplitude du signal. La deuxième partie de l'équation est une fonction de bruit, représentée par  $\mathcal{H}(\eta)$ . Nous considérons que le bruit suit une distribution gaussienne de moyenne  $\mu_K$  et de variance  $\sigma_K^2$ . Une considération de plus en plus conventionnelle dans les systèmes de communication hertziens, du fait qu'elle permet de représenter un environnement très instable en terme de bruit, et favorisant dès lors l'obtention de résultats de performances inférieures à ceux d'un environnement réel.

$$O \left( \frac{\theta}{d_{k,j}}, \alpha \right) = \frac{\theta}{d_{k,j}^\alpha} \quad (5.2)$$

Étant donnée une valeur de détection seuil représentée par  $p_{req}$ , alors la relation de détection entre un capteur  $s_k$  et une cible  $z_j$  est définie comme suit : un capteur peut détecter une cible si 5.3 est satisfaite :

$$P_f \left( O \left( \frac{\theta}{d_{k,j}}, \alpha \right), \mathcal{H}(\eta) \right) \geq p_{req} \quad (5.3)$$

Sinon le capteur ne peut pas détecter la cible. Lorsqu'un capteur peut détecter une cible, la probabilité de détection de  $z_j$  par  $s_k$ , appelée  $p_j^k$ , est exprimée par l'équation 5.4 :

$$p_j^k = Pr \left[ P_f \left( O \left( \frac{\theta}{d_{k,j}}, \alpha \right), \mathcal{H}(\eta) \right) \geq p_{req} \right] = Q(p_{req}) \quad (5.4)$$

Avec  $Pr$  une fonction de probabilité, et  $Q$  la fonction- $Q$ .

Si nous posons :

$$X = \frac{P_f \left( O \left( \frac{\theta}{d_{k,j}}, \alpha \right), \mathcal{H}(\eta) \right) - \mu_K}{\sigma_K} \quad (5.5)$$

Alors

$$Pr[X \geq x] = Q(x) \quad (5.6)$$

avec

$$x = \frac{p_{req} - \mu_K}{\sigma_K} \quad (5.7)$$

Nous obtenons l'équation 5.8 :

$$p_j^k = Pr[X \geq x] = Q\left(\frac{p_{req} - \mu_K}{\sigma_K}\right) \quad (5.8)$$

La fonction de détection est calculée par la fonction  $Q$  définie comme suit :

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp\left(-\frac{u^2}{2}\right) du \quad (5.9)$$

### 5.2.2 Détection collective

Chaque cible peut être détectée par plus d'un capteur. Si une cible est détectée par  $k$  capteurs simultanément, ( $k \geq 1$ ), avec différentes probabilités de détection, la probabilité de détection cumulative à ce point cible, notée  $P_j$  peut être définie comme le produit des probabilités de détection individuelles de tous ses capteurs voisins [1].  $P_j$  est définie par l'équation 5.10.

$$P_j = 1 - \prod_{s_k \in S^j} (1 - p_j^k) \quad (5.10)$$

## 5.3 Formalisation du problème

Étant donné une probabilité de détection seuil  $p_{req}$ , un ensemble  $n$  de capteurs déployés aléatoirement dans un champ  $F$ , et un ensemble de  $m$  de points cibles à surveiller, localisés en des emplacements fixes dans  $F$ , l'objectif du problème de la couverture probabiliste c'est de trouver un ensemble couvrant  $C_j$  satisfaisant les propriétés suivantes :

1.  $|C_{min}|$  est minimal
2.  $\forall z_i \in T, s_k \in S, p_i^k \geq p_{req}$ , pour  $1 \leq i \leq m, 1 \leq k \leq n$

La première propriété indique la minimalité de la cardinalité de l'ensemble extrait. La deuxième contrainte requiert que la probabilité avec laquelle chaque cible est couverte soit au moins égale à la probabilité seuil. Le problème de la couverture d'ensemble probabiliste (MPSC<sup>1</sup>) est un sous-problème du problème de couverture cible évoqué dans le chapitre précédent.

---

1. Minimum Probabilistic Set Cover

**Théorème.** Le problème du MPSC est NP-complet.

**Preuve.** Nous devons d'abord prouver que le problème MPSC appartient à la classe NP.

Considérons un sous-ensemble des capteurs  $S^* \subseteq S$  désigné pour assurer la couverture d'un ensemble fini de cibles. Il serait possible pour un algorithme non déterministe de vérifier en un temps polynomial si chaque cible  $z_j$ , avec  $z_j \in Z$ , est couverte par ses capteurs voisins<sup>2</sup>, avec la probabilité requise  $p_{req}$ . Un exemple d'un tel algorithme non déterministe peut être simulé en effectuant une recherche locale parmi les cibles en fonction de la densité avec laquelle elles sont couvertes (c'est-à-dire le nombre de capteurs capables de fournir une couverture pour cette cible), et tester de manière itérative si la couverture cumulative dépasse ou non la probabilité requise  $p_{req}$ . L'algorithme 7 peut effectuer une telle itération.

Dans un deuxième temps, nous devons prouver que le problème MPSC est NP-difficile.

Pour cela, nous considérons le cas où la probabilité requise est égale à 0. Dans ce cas, la couverture obtenue pour chaque cible est suffisante, peu importe la faible probabilité cumulative avec laquelle chaque cible est couverte. Cela revient à trouver le nombre minimal de capteurs pour couvrir toutes les cibles. Ce qui en fait un problème similaire à celui de l'ensemble couvrant minimal, où tous les éléments de couverture (capteurs) ont le même coût. En limitant le problème à la version de décision du problème de l'ensemble couvrant minimal qui est connu pour être NP-complet, par rapprochement, le problème MPSC est NP-difficile.

En considérant ces deux propriétés montrées ci-dessus, on peut conclure que MPSC est NP-complet.

## 5.4 Formulation PLE

Le problème MPSC pour la couverture de cible peut être formulé en utilisant un modèle de programmation linéaire d'entiers<sup>3</sup>. Pour cela, nous définissons deux variables binaires  $w_k$  et  $q_i^k$  avec  $1 \leq k \leq n$  et  $1 \leq i \leq m$ , pour représenter le fait qu'un capteur appartienne à un ensemble  $C_j$  ou pas :

$$w_k = \begin{cases} 1 & \text{si } s_k \in C_{min} \\ 0 & \text{sinon} \end{cases} \quad (5.11)$$

$$q_i^k = \begin{cases} 1 & \text{si } p_i^k \geq p_{req} \\ 0 & \text{sinon} \end{cases} \quad (5.12)$$

La deuxième variable binaire permet d'exprimer la relation entre la couverture de chaque cible et la probabilité seuil requise ( $p_{req}$ ). Le problème de la couverture de cible probabiliste peut être formulé en utilisant une PLE sous forme d'objectifs et de contraintes comme présentée ci-dessous :

---

2. Les capteurs se trouvant dans le voisinage de ladite cible, et pouvant détecter la cible

3. PLE



Minimiser :  $\sum_{k=1}^n w_k$

Sous Contraintes de :

$$1. 1 - \prod_{k=1}^n (1 - w_k q_i^k p_i^k) \geq p_{req}, 1 \leq i \leq m, 1 \leq k \leq n$$

$$2. w_k = 0 \text{ or } w_k = 1, 1 \leq k \leq n$$

$$3. q_i^k = 1 \text{ or } q_i^k = 0, 1 \leq i \leq m, 1 \leq k \leq n$$

L'objectif du problème c'est de minimiser le nombre de capteurs dans chaque ensemble couvrant extrait, tout en garantissant que chaque cible soit couverte de façon cumulative avec la probabilité requise.

Pour rendre la formulation faisable, des contraintes non linéarisées (comme (1)) doivent être linéarisées. Dans ce cas, le processus de linéarisation peut être effectué en utilisant une transformation logarithmique du produit à la somme, comme montré dans les équations 5.13 et 5.16.

$$1 - \prod_{k=1}^n (1 - w_k q_i^k p_i^k) \geq p_{req} \quad (5.13)$$

$$1 - p_{req} \geq \prod_{k=1}^n (1 - w_k q_i^k p_i^k) \quad (5.14)$$

$$\ln(1 - p_{req}) \geq \ln \left( \prod_{k=1}^n (1 - w_k q_i^k p_i^k) \right) \quad (5.15)$$

$$\ln \left( \prod_{k=1}^n (1 - w_k q_i^k p_i^k) \right) = \sum_{k=1}^n \ln(1 - w_k q_i^k p_i^k) \quad (5.16)$$

Introduisons une nouvelle variable  $y_i^k = \ln(1 - w_k q_i^k p_i^k)$  de façon à ce que les propriétés suivantes soient satisfaites :

i. Si  $w_k = 0$ , alors  $y_i^k = 0$

ii. Si  $w_k = 1$ , alors  $y_i^k = \ln(1 - w_k q_i^k p_i^k)$

La contrainte (1) devient :

$$\sum_{k=1}^n y_i^k \leq \ln(1 - p_{req}) \quad (5.17)$$

La linéarité peut être obtenue en introduisant une nouvelle variable binaire  $u_i^k$  définie telle que :

$$u_i^k = \begin{cases} 1, & \text{si } y_i^k = \ln(1 - q_i^k p_i^k) \\ 0, & \text{sinon} \end{cases} \quad (5.18)$$

Ensuite,  $y_i^k = u_i^k \ln(1 - q_i^k p_i^k)$ . La linéarisation de la contrainte (1) conduit au résultat suivant :

$$\sum_{k=1}^n u_i^k \ln(1 - q_i^k p_i^k) \leq \ln(1 - p_{req}) \quad (5.19)$$

Les propriétés *i* and *ii* sont satisfaites par  $u_i^k$ , i.e. :

- Si  $w_k = 0$ , alors  $u_i^k = 0$
- Si  $w_k = 1$ , alors  $u_i^k = 1$

Dès lors, la relation entre  $w_k$  et  $u_i^k$  peut être exprimée par les équations (5.20) et (5.21) :

$$(1 - w_k) (1 - u_i^k) + w_k u_i^k = 1 \quad (5.20)$$

$$\Rightarrow 2u_i^k w_k - u_i^k - w_k = 0 \quad (5.21)$$

Puisque  $u_i^k w_k$  dans l'équation (5.21) est un produit de deux variables binaires, une nouvelle variable  $v_i^k = u_i^k w_k$  est nécessaire pour sa linéarisation.

Nous introduisons ces trois inégalités suivantes :  $v_i^k = u_i^k w_k \Rightarrow$

1.  $v_i^k \leq u_i^k$ ,
2.  $v_i^k \leq w_k$ ,
3.  $v_i^k \geq u_i^k + w_k - 1$

La PLE résultante pour ce problème est donnée par :

$$\text{Minimiser : } \sum_{k=1}^n w_k$$

Sous Contraintes de :

1.  $\sum_{k=1}^n u_i^k \ln(1 - q_i^k p_i^k) \leq \ln(1 - p_{req})$
2.  $2v_i^k - u_i^k - w_k = 0, 1 \leq i \leq m, 1 \leq k \leq n$
3.  $v_i^k \leq u_i^k, 1 \leq i \leq m, 1 \leq k \leq n$
4.  $v_i^k \leq w_k, 1 \leq i \leq m, 1 \leq k \leq n$
5.  $v_i^k \geq u_i^k + w_k - 1, 1 \leq i \leq m, 1 \leq k \leq n$
6.  $w_k = 0 \text{ or } w_k = 1, 1 \leq i \leq m, 1 \leq k \leq n$
7.  $q_i^k = 0 \text{ or } q_i^k = 1, 1 \leq i \leq m, 1 \leq k \leq n$
8.  $u_i^k = 0 \text{ or } u_i^k = 1, 1 \leq i \leq m, 1 \leq k \leq n$
9.  $v_i^k = 0 \text{ or } v_i^k = 1, 1 \leq i \leq m, 1 \leq k \leq n$

Ce qui nous emmène à une formulation linéaire à 5 variables. Il est possible d'adopter l'une des méthodes classiques d'approches de résolution de problème linéaire, tels que la méthode

des plans sécants et le principe de séparation-évaluation<sup>4</sup> pour résoudre de manière exacte la PLE. Cependant, l'idée d'une résolution exacte étant à peu près discréditée du fait des éventuelles difficultés en termes de faisabilité et de temps d'exécution, nous nous rabattons sur une approche d'approximation susceptible de converger vers un résultat acceptable en un temps raisonnable. Dans ce qui suit, nous allons formuler deux solutions algorithmiques basées sur une stratégie de recherche local, et un algorithme glouton pour résoudre efficacement ce problème.

## 5.5 Approximations basées sur une recherche locale

---

### Algorithme 3 : Recherche Local d'Ensembles Couvrant

---

```

entrée :  $S_0, Z_0, E_{S_0}, p_{req}$ 
sortie :  $C$ 

Collecter l'énergie restante  $e_{rem}^k$  de chaque capteur  $s_k$ ; 1
Appliquer Tri-rapide en fonction de  $e_{rem}^k$ ; 2
 $C = \{\}$ ; 3
foreach  $s_k \in S_{sorted}$  do 4
     $z_k$  = sous-ensembles de cibles couvertes; 5
     $s_{neigh}^k$  = sous-ensemble des capteurs couvrant  $z_k$ ; 6
     $z_{other}$  = union des sous-ensembles de cibles couvertes par  $s_{neigh}^k$ ; 7
     $z_{overlap} = z_{other} \cap z_k$ ; 8
     $estRedundant(s_k) = False$ ; 9
    foreach  $z_i \in z_{overlap}$  do 10
        Calculer  $P_i$  avec l'équation 5.10; 11
        if  $P_i \leq p_{req}$  then 12
             $estRedundant(s_k) = True$ ; 13
        end 14
    end 15
    if  $estRedundant(s_k)$  then 16
         $SLEEP(s_k)$ ; 17
    else 18
         $ACTIVE(s_k)$ ; 19
         $C = C \cup s_k$ ; 20
    end 21
end 22
return  $C$ ; 23

```

---

Nous introduisons d'abord un algorithme de recherche locale avec les même objectifs que ceux formulés dans la PLE, c'est-à-dire assurer la couverture par rapport à la probabilité  $p_{req}$  tout en tentant de minimiser le mieux possible le nombre de capteurs sélectionnés. Le pseudo-code de l'algorithme est présenté dans 7. Afin d'effectuer la recherche locale, l'algorithme commence par collecter l'énergie restante  $e_{rem}^k$  de tous les capteurs disponibles, puis applique un tri rapide afin d'organiser les capteurs dans l'ordre croissant en fonction de  $e_{rem}^k$ . L'entrée de l'algorithme

---

4. branch and bound

de recherche locale est l'ensemble des capteurs disponibles aux niveaux d'énergie résiduels non nuls. En commençant par le capteur ayant la plus faible quantité d'énergie, et en remontant vers celui avec le plus haut niveau d'énergie résiduel, l'algorithme calcule, pour chaque capteur, le sous-ensemble de cibles couvertes  $z_k$  et le sous-ensemble des cibles  $z_{other}$  couvertes par ses capteurs voisins  $s_{neigh}^k$ , et en dérive le sous-ensemble de cibles pouvant être couvertes par les capteurs dans  $s_{neigh}^k$ .

Afin de décider si le capteur  $s_k$  doit passer en état de veille ou pas, une règle d'éligibilité est appliquée. L'idée derrière cette règle est basée sur le fait qu'un capteur  $s_k$  peut être programmé pour entrer en mode inactif pendant une période si ses capteurs voisins sont en mesure de fournir la couverture nécessaire pour chaque cible en  $z_k$ . Pour prendre une telle décision, l'algorithme vérifie, pour chaque cible,  $z_i \in Z_k$  si la probabilité de détection cumulative fournie par les capteurs voisins est supérieure ou égale à  $p_{req}$ . Si tel est le cas, ce capteur peut être inactif, en supposant qu'il restera assez de capteurs de garde dans la liste non encore visitée des capteurs, avec la rentabilité requise pour couvrir ses cibles voisines.

## 5.6 Approximation basée sur une stratégie gloutonne

### 5.6.1 Ensembles disjoints et non-disjoints

L'algorithme que nous proposons est basé sur une version modifiée de la couverture d'ensembles gloutonne. Lors de la construction d'un ensemble couvrant, l'approche classique sélectionne itérativement le capteur couvrant le plus grand nombre de cibles non couvertes, supprime ces cibles de la liste des cibles non couvertes et répète le même processus jusqu'à ce que la liste des cibles non couvertes soit vide. Le sous-ensemble minimum de capteurs, pouvant couvrir une cible, place une limite sur le nombre d'ensembles calculables pour une solution optimale. Cette limite est appelée le maximum théorique. Dans ce cas, cette cible représente la cible la moins couverte, que nous appelons ici cible critique. Formellement, si  $z_t$  est la cible critique, alors  $|S^t|$  représente le maximum théorique dans le cas de la couverture d'ensembles disjoints. Ceci est dû au fait que chaque cible doit être couverte par au moins un capteur dans un ensemble couvrant, et qu'un capteur peut faire partie d'au plus un ensemble.

Lorsque les ensembles produits sont non-disjoints, un capteur peut être utilisé au plus  $d$  fois, avec  $d \geq 1$ , pour faire partie des ensembles. Le maximum théorique sera égal à  $d|S^t|$ . Pour calculer le nombre d'ensembles le plus proche du maximum théorique, il faut adopter une stratégie de sélection efficace des capteurs, qui évite d'inclure plus d'un capteur couvrant une cible critique dans un ensemble. Ceci, peut être réalisé en concevant une fonction de profit qui représente la contribution de chaque capteur. Cette fonction est appliquée à chaque étape de sélection de capteur, et le capteur ayant la plus grande valeur pour cette fonction est sélectionné.

### 5.6.2 Stratégie de sélection gloutonne

Le déploiement étant aléatoire, toutes les cibles ne sont pas entourées par le même nombre de capteurs. Chaque capteur a une contribution lors de la sélection. Celle-ci dépend du nombre de cibles sous sa portée et la probabilité de détection de ces cibles. La fonction de profit, utilisée ici pour rendre commode la sélection des capteurs, tient compte de l'énergie restante de chaque capteur, qui représente le poids qui lui est associé. Nous définissons pour chaque capteur  $s_k$  et cible  $z_j$  la fonction modélisant la relation de couverture (CRF) représentée par  $w_j^k$  dans l'équation (5.22).

$$w_j^k = \frac{p_j^k}{\sum_{s_{k'} \in Z_0} p_j^{k'} \frac{E^k}{E^{k'}}} \quad (5.22)$$

La CRF entre  $s_k$  et  $z_j$  est égale au rapport entre la probabilité de détection de  $s_k$  et  $z_j$  et la somme des probabilités de détection de tous les capteurs sis à la portée de  $z_j$ . Si ces derniers peuvent surveiller  $z_j$  avec une bonne couverture, cela implique que  $z_j$  est densément couverte, avec une plus faible CRF pour  $s_k$  par rapport au cas où  $z_j$  est faiblement couverte. De plus,  $w_j^k$  est égal à 0 si  $z_j$  ne peut être surveillée par aucun capteur. La plus grande valeur correspond au cas où  $z_j$  ne peut être surveillée que par un seul capteur.

En se basant sur l'équation 5.22, nous définissons dans 5.23 la contribution totale de  $s_k$  considérant  $Z_0$ , lequel représente sa fonction de profit, que nous désignons par  $w_{Z_0}^k$ .

$$w_{Z_0}^k = \sum_{z_j \in Z_0} \frac{p_j^k}{m E^k \sum_{s_{k'} \in S_0} \frac{p_j^{k'}}{E^{k'}}} = \frac{1}{m} \sum_{z_j \in Z_0} w_j^k \quad (5.23)$$

Avec  $E^k$  l'énergie du capteur  $k$ ,  $m$  le nombre de cibles et  $Z_0$  l'ensemble des cibles. La fonction de profit de  $s_k$  par rapport à  $Z_0$  est égale à la somme normalisée des CRF entre  $s_k$  et toutes les cibles qui appartiennent à  $Z_0$ . Étant donné que la détection des cibles est probabiliste, la sélection des capteurs à inclure dans les ensembles couvrants doit d'abord prendre en considération la quantité de cibles critiques qu'ils peuvent surveiller, la proportion de cibles non couvertes qu'ils peuvent surveiller, et le niveau de détection des cibles. La fonction de profit décrite dans l'équation 5.23 permet de favoriser les capteurs qui couvrent le plus grand nombre de cibles non couvertes avec une probabilité de détection élevée, sans perdre de vue l'exigence de gérer les cibles critiques.

En guise de remarque, lorsqu'un modèle booléen est considéré, de nombreux capteurs peuvent avoir la même valeur de fonction de profit, alors qu'ils ne couvrent pas leurs cibles voisines de la même manière. La fonction de profit probabiliste permet plus aisément d'effectuer un choix, en distinguant ceux aux probabilités de détection plus élevées que d'autres.

**Algorithme 4 : ALPGEC : ALgorithme Probabiliste Glouton d'Ensembles Couvrants**


---

```

input   :  $S_0, Z_0$ 
output :  $C$ 

 $Z_{cur} = Z_0;$                                      1
 $S_{cur} = S_0;$                                      2
 $C = \phi;$                                            3
while  $Z_{cur} \neq \phi$  do                             4
     $maxCE = 0 ;$                                        5
     $s_{selected} = vide;$                              6
    foreach  $s_k \in S_{cur}$  do                         7
        Calculer la fonction de profit avec l'équation 5.23; 8
        if  $ce(s_k) \geq maxCE$  then                 9
             $s_{selected} = s_k;$                    10
             $maxCE = ce(s_k);$                        11
        end                                           12
    end                                               13
    Supprimer les cibles couvertes par  $s_{selected}$  de  $Z_{cur}$ ; 14
    Ajouter  $s_{selected}$  à l'ensemble courant;          15
end                                                 16
while  $\forall z_k, z_k \text{ not } p_{th}\text{-covered}$  do        17
    foreach  $s_k \in S_{cur}$  do                       18
        if  $p_{other}^k \leq p_{req}$  then                19
             $C = C \cup s_k;$                        20
        end                                           21
    end                                               22
end                                                 23
return  $C;$                                          24

```

---

### 5.6.3 Présentation de l'algorithme glouton

L'algorithme est présenté sous forme de pseudo-code dans l'Algorithme 4. Cette seconde solution a pour objectif de garantir la couverture de toutes les cibles, tout en essayant de sélectionner un nombre minimal possible de capteurs, et puis de garantir que chaque cible soit couverte avec la probabilité requise  $p_{req}$ . Pour ce faire, l'algorithme est composé de deux phases principales, à savoir la phase de génération des ensembles couvrants, et la phase de  $p_{req}$ -couverture. Lors de la génération des ensembles couvrants, l'algorithme détient une liste des cibles non couvertes initialement égale à l'ensemble initial de cibles  $Z_0$ , et sélectionne avidement le capteur ayant la rentabilité maximale à chaque itération. Ensuite, l'algorithme supprime le sous-ensemble des cibles couvertes à partir de  $Z_{cur}$ , et réitère le processus sur l'ensemble des cibles non encore couvertes jusqu'à ce que toutes les cibles soient couvertes.

Afin de garantir que chaque cible soit  $p_{req}$ -couverte, la deuxième phase de l'algorithme est appliquée. A ce niveau, l'algorithme détient la liste des cibles pas encore  $p_{req}$ -couvertes et procède par élimination, tout en ajoutant à l'ensemble couvrant extrait le nombre requis de capteurs pour atteindre une  $p_{req}$ -couverture.

## 5.7 Évaluation de performances

Dans cette section, nous évaluons les performances des algorithmes en termes de détection de cibles et de nombre d'ensembles couvrants. Nous simulons et puis effectuons une comparaison avec d'autres travaux de la littérature [78, 15, 14, 101, 102]. Étant donné qu'aucun de ces travaux ne considère un modèle de couverture probabiliste, il est plus pertinent de comparer notre solution à chacun de ces travaux en termes de nombre d'ensembles couvrants calculés.

### 5.7.1 Environnement de simulation

Les simulations que nous présentons dans cette section ont été réalisées sur un environnement Matlab 7.02. Nous considérons un espace à deux dimensions de 100x100 unités de surface représentant la zone de déploiement. Les cibles sont d'abord déployées dans la zone, suivies d'un déploiement aléatoire de capteurs de manière à apporter une redondance dans le déploiement. Les positions des capteurs et des cibles sont créées en générant des coordonnées aléatoires, suivant une distribution uniforme. Puisque la distribution des capteurs dans le champ de détection suit une distribution uniforme et que nous traitons essentiellement du problème de gestion de la couverture de cible, l'effet aléatoire sur les valeurs de détection des couvertures est négligé dans l'interprétation de nos résultats de simulation. Les paramètres numériques de simulation sont présentés dans le tableau 5.1.

TABLE 5.1 – Paramètres de simulation.

Paramètres	Value
Dimension de terrain	100m X 100m
Nombre de capteurs	10, ..., 100
Nombre de cibles	10
Portée maximale	30m
Energie des capteurs	1j
Temps d'exécution d'un EC (rt)	1/2

TABLE 5.2 – Valeurs de détection minimales, maximales et moyennes après 100 instances d'exécutions.

Nb.Capteurs	Ensembles Disjoints			Ensembles Non-disjoints		
	Min.	Max.	AVG	Min.	Max.	AVG
30	0,055	0,241	0,128	0,083	0,279	0,151
40	0,046	0,282	0,133	0,073	0,290	0,151
50	0,036	0,266	0,121	0,074	0,305	0,161
60	0,039	0,247	0,116	0,070	0,312	0,153
70	0,036	0,260	0,120	0,068	0,332	0,158
80	0,036	0,302	0,125	0,060	0,327	0,151
90	0,031	0,289	0,119	0,060	0,355	0,159
100	0,037	0,314	0,125	0,058	0,382	0,155
Écart-type ( $\sigma$ )	7,7E-03	2,6E-02	5,5E-03	8,7E-03	3,4E-02	3,8E-03

### 5.7.2 Résultats de simulation

#### Stratégie de sélection des capteurs

Lors de la sélection des capteurs à inclure dans les ensembles couvrants, l'algorithme de recherche local commence par trier les capteurs en fonction de leurs réserves d'énergie  $e_{rem}^k$ . Les capteurs aux  $e_{rem}^k$  plus petits sont sélectionnés en premier. Ce qui permet de maintenir une stabilité tout au long de la durée de vie du réseau. Ce maintien est dû à la stratégie de sélection discriminant les capteurs n'ayant pas encore utilisé leurs sources d'énergie, au détriment de capteurs déjà sélectionnés avec un niveau d'énergie plus faible. L'algorithme glouton sélectionne à chaque itération le capteur présentant le profit maximal. De cette façon, les capteurs présentant les meilleures rentabilités, en fonction de la stratégie visée dans l'action gloutonne composent le sous-ensemble couvrant. A celui-ci, sont ajoutés d'autres capteurs jusqu'à ce que toutes les cibles soient couvertes avec la probabilité requise.

#### Niveaux de détection des cibles

Les mesures numériques présentées dans le tableau 5.2 décrivent les résultats de la simulation lorsque l'on fait varier le nombre de capteurs de 30 à 100 et que le nombre de cibles est fixé à 10 et le rayon de capture à 30 m. Nous déroulons la simulation en exécutant 100 fois l'algorithme glouton, et en considérant un nouveau déploiement pour chaque valeur de nombre de capteurs,



TABLE 5.3 – Nombre d’ensembles couvrants disjoints comparés à Slijepcevic[78], Zorbas (B{GoP}) et WGA[30].

Nb.Capteurs	Ensembles Disjoints			
	Slijepcevic	B{GoP}	WGA	Disjoint ALPGEC
30	6,6	8,8	8,6	9,2
40	6,9	9,2	9,3	11,8
50	8,5	11,4	12,3	14,6
60	10,5	12,9	13	13,6
70	12,5	17,1	18	21,8
80	16,2	23,4	24,4	25,2
90	17,8	26,8	28,4	28,1
100	18	27,6	28,2	29,4

TABLE 5.4 – Nombre d’ensembles couvrants non-disjoints comparés à [102]) et Cardei [15].

Nb.Capteurs	Ensembles non-disjoints		
	Non-disjoint B{GoP}	Cardei	Non-disjoint ALPGEC
30	16,4	16,4	12,8
40	20,2	19,2	19,2
50	24,3	23	18,4
60	28,3	28	26,8
70	33,7	32,8	32,8
80	41,9	41,6	37,8
90	48,5	48,2	43,6
100	58,4	58,2	47

puis nous estimons les valeurs de détection minimum et maximum en considérant la moyenne arithmétique des valeurs calculées.

En considérant les valeurs numériques du tableau 5.2, nous pouvons remarquer une légère dégradation des valeurs de détection minimales lorsque le nombre de capteurs varie de 30 à 100 et en parallèle une légère amélioration des valeurs de détection maximales, pour l’algorithme glouton disjoint et non-disjoint. Ceci peut s’expliquer par l’hypothèse selon laquelle lorsque le nombre de capteurs augmente, cela se traduit par plus de composants dans le champ de détection qui conduisent à une certaine redondance dans la couverture des cibles. Cela implique une plus grande probabilité que chaque cible soit couverte de manière redondante, puis qu’elle soit détectée par ses capteurs voisins avec une valeur de détection plus élevée, contrastant avec le cas où les cibles seraient moins couvertes. L’amélioration de la détection des cibles peut conduire à la dégradation des valeurs de détection minimales lorsque le nombre de capteurs augmente. A partir de l’observation de l’écart-type après 100 exécutions de l’algorithme avec différentes configurations, on peut considérer que les valeurs de détection maximales et minimales présentées dans le tableau 5.2 sont assez représentatives.

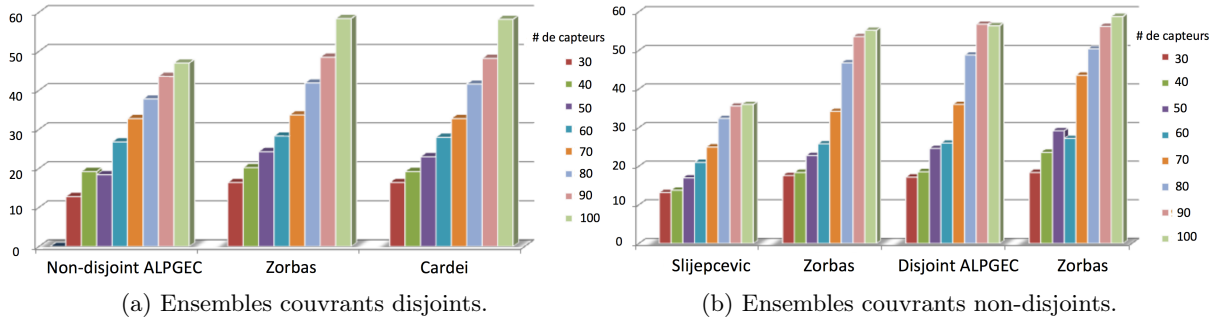


FIGURE 5.1 – Histogrammes du nombre d'ensembles couvrants et comparaisons.

### Calcul d'ensembles couvrants

Nous comparons les nombres d'ensembles couvrants disjoints obtenues à B GoP proposé par Zorbas et al.[101] et MCMC-H (Slijepcevic et al.) [78]. Les ensembles non disjoints sont comparés aux solutions proposées dans Cardei et al.[15] et Zorbas et al.[102]. Les mêmes paramètres du que précédemment sont considérés et la simulation est déroulée 20 fois, puis la valeur moyenne des ensembles couvrants est calculée pour chaque nombre de capteurs. Les résultats numériques sont présentés dans les tableaux 5.3 et 5.4 et représentés sous forme d'histogrammes sur la figure 5.1. En considérant les ensembles disjoints, nous pouvons remarquer en observant les résultats numériques du tableau 5.3, représentés sur la figure 5.1a, que pour chaque instance de nombre de capteurs, les ensembles couvrants disjoints montrent les meilleures performances en terme d'ensembles calculées, comparé aux autres algorithmes. En observant les algorithmes non disjoints (figure 5.1b), les résultats de la simulation sont dominés par des performances acceptables, en considérant le nombre d'ensembles couvrants calculées.

Une remarque générale est que la cible critique est bien gérée par la fonction de profit lors de la sélection des capteurs. Lors du calcul des ensembles disjoints, chaque capteur ne peut pas faire partie de plus d'un ensemble. Cela signifie que pendant l'activation des ensembles couvrants, chaque capteur est activé jusqu'à épuisement total des batteries. L'énergie de tous les capteurs vivants est donc la même si nous négligeons les petites dépenses d'énergie pendant les périodes où les capteurs fonctionnent en mode veille. Dans ce cas, la fonction de profit devient plus précise car le rapport entre l'énergie de deux capteurs lors de l'évaluation du coût des capteurs est simplifié à 1. Sachant que l'algorithme glouton tente de satisfaire les objectifs suivants : garantir la couverture de toutes les cibles avec la probabilité requise  $p_{req}$ , tout en essayant de sélectionner un nombre minimal possible de capteurs, la fonction coût-efficacité que nous appliquons ici à la différence de celles utilisées dans notre travail précédent et les autres algorithmes auxquels nous nous référons dans notre comparaison, doit favoriser les capteurs non seulement en fonction de leur contribution au proportion de cibles couvertes, mais doit également prendre en compte une probabilité de détection associée à chaque relation de couverture capteur-cible. Lors de chaque tour de sélection des capteurs, le meilleur capteur à sélectionner couvre le plus grand nombre

de cibles avec la probabilité de détection la plus élevée. Ainsi, nous pouvons affirmer que notre stratégie de sélection des capteurs pour construire des ensembles couvrants, fournit des résultats acceptables sur le nombre d'ensembles couvrants, ainsi que sur la façon dont chaque cible est détectée.

## 5.8 Conclusion

Ce chapitre a été consacré au problème de la gestion de la couverture cible lorsqu'une détection probabiliste est considérée. Lorsque les capteurs sont déployés de manière aléatoire et redondante dans une zone pour surveiller un ensemble de cibles sises à des emplacements connus, et que la détection des cibles est probabiliste, comment pouvons-nous partitionner l'ensemble des capteurs disponibles en un nombre maximal d'ensembles couvrants, de sorte que chaque ensemble puisse garantir une couverture complète des cibles ? Après avoir formulé le problème à l'aide d'une programmation linéaire d'entier, nous avons proposé des algorithmes approximatifs, qui tentent d'organiser les capteurs en un nombre maximum d'ensembles couvrants. Ces ensembles seront activés successivement, jusqu'à ce que tous les capteurs épuisent leur énergie. De bonnes performances ont été mises en évidence, en comparaison à d'autres solutions dans la littérature. La solution est cependant sous-optimale, sans rapport d'approximation. Également, le problème de la connectivité des nœuds n'a pas été abordé dans ce document, par conséquent les capteurs sont censés être directement connectés à un nœud collecteur pour reporter des informations collectées.

