

SORBONNE UNIVERSITÉ



Dissertation sur les algorithmes présentés
dans [Li, Thai, Wang, Yi, Wan, and Du.
Wireless Communications and Mobile
Computing, 2005].

Heythem NOUARI

8 novembre 2020

Table des matières

1	Introduction	2
2	Présentation De La Solution	2
2.1	Ensemble Dominant	2
2.2	Ensemble Indépendant Maximum (MIS)	3
2.3	Ensemble Dominant Connexe (CDS)	4
2.4	L'algorithme S-MIS	4
3	Implémentation	4
4	Conclusion	5
5	Bibliographie	6

1 Introduction

Le développement des réseaux de télécommunications modernes ces derniers temps a provoqué une importante évolution dans les moyens de correspondances. Chose qui a nécessité une fort déploiement d'antennes-relais et qui ne cesse de grandir. le transfert de l'information entre les antennes crée donc des problématiques de coût et d'efficacité et pousse les chercheurs à chercher des solutions pour assurer ce transfert d'informations en utilisant un nombre minimal d'antennes. Ce réseau peut être représenté par un graphe où les noeuds sont les antennes. Trouver une solution pour avoir un graphe connexe à partir de ce graphe en utilisant un nombre minimal de noeuds et dans un temps d'exécution optimale nous aidera à résoudre la problématique. L'article qu'on va étudier propose une solution expérimentale à ce problème.

Dans ce rapport nous allons d'abord présenter les principes de déduction d'un graphe connexe optimisé puis présenter et analyser les algorithmes d'optimisations proposés dans cet article

2 Présentation De La Solution

Nous allons définir les graphes et termes liés au concept d'un ensemble dominant connexe optimal, puis nous analyserons les algorithmes décrit dans l'article, pour construire une implémentation efficace.

2.1 Ensemble Dominant

Un ensemble dominant (DS) est un sous ensemble de points tel que chaque point du graphe appartient soit à l'ensemble dominant soit il est voisin d'un des point de l'ensemble dominant. Pour obtenir un DS à partir d'un graph $G=(V,E)$ nous avons procédé ainsi :

- un algorithme glouton qui rajoute les sommets qui ont le maximum de voisins , supprime les voisins de ces points et relance cette opération jusqu'à la fin des points.
- on améliore cet le glouton en prenant son résultat puis pour chaque point du graphe p on teste si en remplaçant un couple de points (a,b) dans le DS notre on obtient un autre DS . et cette technique diminuera la taille du DS résultat.

Data : points : liste des sommets

Result : DS : l'ensemble dominant

```
1 while points != vide do
2   u=maxConnected(points)
3   DS.add(u)
4   points.removeAll(neighbors(u))
5   points.remove(u)
6 end
```

Algorithme 1 : l'algorithme DS-Glouton

```

Data : points : liste des sommets
Result : DS : l'ensemble dominant
1 ancien =DS-Glouton(points)
2 courant=new Point[]
3 while courant.size() < ancien.size() do
4   |   ancien=courant courant=optimiser(ancien)
      |   // optimiser pour chaque point p de points , prend un couple du DS et le
      |   remplace par p si le resultat est un DS on prend ce nouveau DS
5 end

```

Algorithme 2 : L'algorithme avec l'optimisation

2.2 Ensemble Indépendant Maximum (MIS)

```

Data : points : liste des sommets
Result : MIS : Ensemble indépendant Maximum
1 for(p :points) p.color=WHITE
2 s=points.get(0)
3 s.color=BLACK
4 MIS.add(s)
5 for(u :neighbor(s)) u.setActif(true)
6 while existWhite(points) do
7   |   actifMax = un noeud actif avec max de voisin de couleur white
8   |   actifMax.color=BLACK
9   |   MIS.add(actifMax)
10  |   for(u :neighbor(actifMax)) u.color=GREY
11  |   for(v :neighbor(u)) v.setActive(true)
12 end

```

Algorithme 3 : L'algorithme MIS

Un ensemble indépendant maximum d'un graphe est un sous ensemble de points tel que entre chaque paire de points du MIS il n'existe pas d'arrête . On remarque que cet ensemble est un ensemble dominant, mais par forcément le plus optimale. pour construire un MIS de manière gloutonne : on ajoute un premier point quelconque , puis un deuxième point en le choisissant parmi les non adjacents du premier et ainsi de suite jusqu'à ce qu'on puisse plus trouver de point. remarque : Dans l'article Li et Al utilisent un MIS qui pour chaque paire de point doit avoir une distance de deux arrêtes.

2.3 Ensemble Dominant Connexe (CDS)

Un ensemble dominant connexe (CDS) est un ensemble dominant dans lequel il existe un chemin (ensemble d'arretes) entre n'importe quel point A et n'importe quel point B. Les chercheurs dans cet article ont proposé Deux algorithmes pour obtenir un CDS d'un graphe :

- Un premier algorithme qui construit le CDS noeud par noeud, on prend arbitrairement un premier noeud, qui sera notre premier point dominant, que l'on colorie en noir, puis on colorie tous les points dominés en gris(voisins du point pris au départ), et on réitère jusqu'à ne plus avoir de points blancs.
- Un deuxième algorithme qui identifie tous les points dominants, puis les relie entre eux avec des noeuds intermédiaire pour former un ensemble de point dominants connexe.

2.4 L'algorithme S-MIS

cet algorithme est une nouvelle approche gloutonne du problème. La première étape de cet algorithme est de construire un MIS où chaque noeud a au plus 5 noeud voisins indépendant. Dans le graphe résultant du MIS, chaque point a une distance de deux intervalles avec son points voisin. Puis, dans la seconde étape , on connecte tous les noeuds de notre MIS pour former un CDS (car par définition un MIS est un ensemble dominant). L'algorithme permettant de rendre le MIS connexe repose sur une coloration des points : tous les points du MIS sont initialement noir, tous les autres sont gris, et deviennent bleu s'ils ont n des voisins noir, voisins qui doivent être issus de différentes composantes connexes (n allant de 5 à 2).

Remarque : Dans l'algorithme ci dessus, l'article propose d'ignorer les connexions entre les noeuds bleus au moment du choix du connecteur pour un noeud noir , et cela n'affectera presque pas le résultat.

3 Implémentation

Nous avons pu implémenter l'algo S-MIS (la fonction algoArticle).voici ce que retourne en l'appliquant sur le graphe fourni dans le canvas du TME3 .

Clavier:

- 'r' pour rafraîchir le nuage de points
- 'c' pour lancer calculConnectedDominatingSet
- 'h', 'j', 'k', 'l' pour déplacer les points

Temps de calcul: 362 ms

Taille de l'ensemble: 80

Connected Dominating Set Viewer

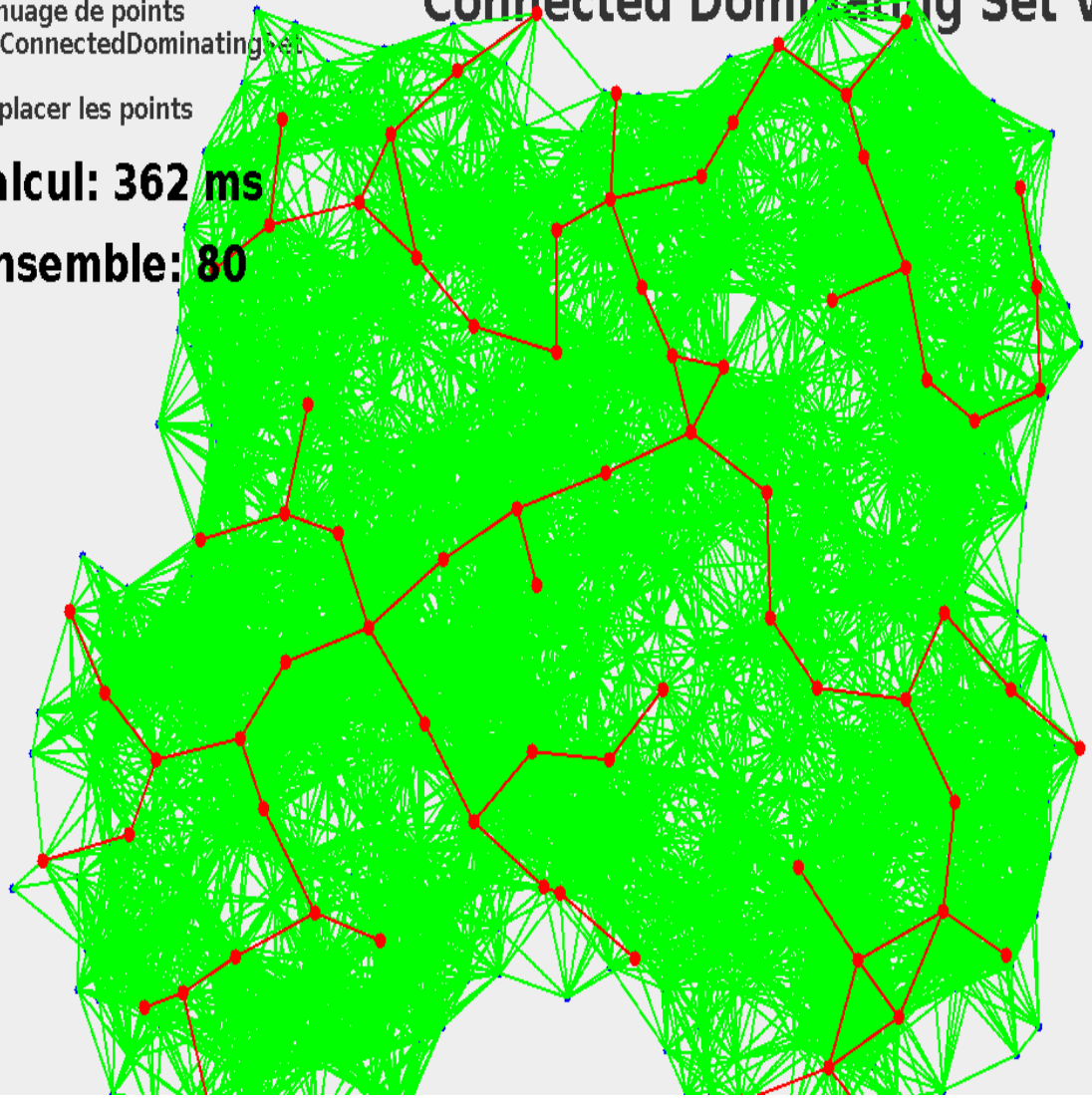


FIGURE 1 – résultat de l’algo S-MIS

4 Conclusion

Malheureusement Par manque de temps je n’ai pas pu faire la comparaison entre l’algorithme S-MIS et l’approche du tme3 (l’utilisation de l’arbre de Steiner) , néanmoins le temps d’exécution de l’algo S-MIS est clairement très petit par rapport à l’approche vu en tme3 .Chose que j’ai pu vérifier dans les expériences des étudiants des années précédentes.

La partie la plus dure était sans doute l’implémentation de l’Algo S-MIS.mais j’imagine que l’implémentation de l’arbre de Steiner aurait pris aussi un temps considérable.

5 Bibliographie

[Yu et al. Connected dominating sets in wireless ad hoc and sensor networks – A comprehensive survey, 2013]

Dissertation sur les algorithmes présentés dans [Li, Thai, Wang, Yi, Wan, and Du. Wireless Communications and Mobile Computing, 2005].