



**POLYTECHNIQUE
MONTRÉAL**

**LE GÉNIE
EN PREMIÈRE CLASSE**

POLYTECHNIQUE DE MONTRÉAL

INF4215

INTRODUCTION À L'INTELLIGENCE ARTIFICIELLE

Laboratoire #1

Auteur:

Slimane BENDELLOUL - 1643556

Mai-Khanh LE - 1571365

Remis à:

Mathieu BUCHER

Le 14 Février 2016

Contents

Présentation de la structure	2
État:	2
Antenne:	2
Cercle englobant:	2
Recherche Arborescence:	2
Recherche locale:	2
Question 1	2
Question 2	2
Recherche par arborescence:	2
Recherche locale:	3
Amélioration générale:	3
Algorithme à choisir	3

Présentation de la structure

État:

Nous avons imaginé notre état comme l'ensemble d'antennes présentes ainsi que l'ensemble des positions non couvertes.

Antenne:

L'antenne est représentée par ses coordonnées, soit les points x-y ainsi que le rayon. De plus, la liste des points couverts y présente afin de pouvoir vérifier la couverture de tous les points

Cercle englobant:

Pour cette partie, nous avons décidé d'utiliser un module déjà existant que nous avons intégré à notre projet. La solution est en $O(n)$ ce qui est très avantageux en temps de calcul. Il se situe dans le fichier "smallesenclosingcircle.py"

Recherche Arborescence:

Nous avons opté par une recherche à coût optimale. Nous avons utilisé notre propre implémentation afin d'avoir plus de contrôle, mais aussi afin d'avoir une certaine appropriation de notre projet. Nous avons utilisé une PriorityQueue au lieu de la liste proposé dans le cadre du cours. La Queue contient un doublet;cout,état; afin d'avoir l'état le moins cher en premier, vu que cette dernière est ordonnée du plus petit au plus grand.

Recherche locale:

Nous avons décidé d'utiliser un recuit simulé pour ce problème, car nous avons jugé son comportement plus optimal. Les paramètres(température, coef, ont été spécialement mis à jour pour le problème type.

Question 1

Cette fonction fait un filtrage des éléments ne correspondant pas au critère de sélection. Une application serait d'appliquer cette fonction à nos états pour avoir seulement des états qu'on jugera valide!

Question 2

Recherche par arborescence:

.

Notre approche pour la recherche par arborescence laisse à désirer. En effet, bien que nous soyons certains de bien avoir défini nos états, notre conception d'états initiaux et de mouvements reste à désirer!

Au tout début, nous avons commencé par un état initial composé d'une antenne englobant tous les éléments. Nous avons par la suite 3 actions, retirer un point à une antenne, ajouter un point à une antenne et créer une nouvelle antenne avec un des points vacants. Notre état final correspondait en la présence d'un état qui ne contenait aucune antenne. Nous aurions ainsi parcouru tous les états possibles. Cela prenait cependant extrêmement beaucoup de temps, malgré le fait que nous avons mis une vérification en hachage sur le cout.

Nous avons alors décidé de changer de stratégie. Maintenant, nous commençons avec une solution où l'état initial correspond à une configuration vide. Nous avons aussi changé d'actions. Nous n'enlevons plus de points de nos antennes, ce qui réduit considérablement le nombre d'états à considérer. Notre but est maintenant d'atteindre une couverture maximale.

Cette approche est plus rapide, cependant, il se peut qu'on arrive à une solution maximale qui ne soit pas optimale(théoriquement), nous n'avons pas eu ce problème lors de nos tests, mais cela est une hypothèse à considérer.

Recherche locale:

Nous avons décidé de choisir une méta-heuristique pour l'approche en recherche locale. En effet, cette approche donne une solution optimale la plupart du temps avec une certaine facilité d'implémentation. C'est notre stratégie de recherche de voisin qui a été la clé de la réussite de cette implémentation

Notre état initial correspond à autant d'antennes que de points. Nous évitons ainsi d'implémenter une méthode vorace pour trouver la solution de départ, car nous la jugeons un bon point de départ. Par la suite, nous effectuons une recherche par voisinage en choisissant au hasard une antenne et nous lui enlevons au hasard un de ces points. Ensuite, nous ajoutons cette position à une des antennes encore prise au hasard. Cela nous permet de trouver la solution optimale en un temps rapide!

Nous aurions cependant dû utiliser une recherche tabou, car nous avons la possibilité de catégoriser les états par un hachage(le cout) et d'établir une liste d'actions. Cette approche est plus optimale qu'un recuit simulé, car notre implémentation nous retourne le mauvais résultat 1 fois sur 10.

Amélioration générale:

Nous avons délibérément ignoré un état un particulier, soit un état où la meilleure solution contiendrait des antennes où des antennes peuvent couvrir des points en commun. Nous n'avons pas pu trouver un exemplaire où cette situation serait la meilleure solution, mais cela n'empêche que cela serait une bonne addition. Seulement voilà, une telle implémentation s'est révélée ardue

Algorithme à choisir

Nous croyons que l'algorithme à choisir dépend du type d'exemplaire. En effet, pour un exemplaire avec peu de points, une recherche par arborescence est à préférer. Choisir une recherche locale ne serait pas judicieux, car les itérations ne dépendent pas des états à visiter, mais de la température de l'algorithme(dans notre cas, le recuit simulé). Nous allons assumer que la taille de l'exemplaire sera tout au plus de taille moyenne, donc nous préférons soumettre l'algorithme de recherche par arborescence.