

Travail pratique #1

RAPHAEL LAPIERRE 1644671
ALEXANDRE ST-ONGE MATRICULE 1623576
École Polytechnique de Montréal

Dans le cadre du cours
INF4215 - Introduction à l'intelligence artificielle

14 Février 2016

Explication des algorithmes

Recherche en arbre

La recherche en arbre commence avec l'état initial vide et n enfants avec chacun 1 antenne qui couvre une position, où n est le nombre de place à couvrir. Par la suite, l'enfant d'un état est calculé en trouvant le point non couvert le plus près de la dernière antenne placé et en couvrant ce dernier soit en ajoutant une nouvelle antenne ou en agrandissant la dernière antenne placé. L'algorithme de Dijkstra est utilisé pour parcourir l'arbre d'état afin de trouver une solution.

Recherche locale

Notre algorithme de recherche locale effectue un recuit simulé à partir d'une solution obtenue avec une algorithme vorace naïf. Le recuit simulé fonctionne en échangeant des points entre les antennes.

Questions

Question 1

Expliquez ce que fait cette fonction et fournissez un exemple utilisant cette fonction.

```
def fct(predList, inputList):  
    return filter(lambda x: all([f(x) for f in predList]),  
                  inputList)
```

La fonction filtre la liste `inputList` pour ne retourner que les éléments qui respectent tout les prédicats présents dans `predList`. Si `inputList` est une liste d'état, il serait donc possible d'utiliser cette fonction pour filtrer la liste afin d'obtenir seulement que les états respectant tout nos prédicats.

Question 2

Recherche en arbre

Le point fort de la recherche en arbre que nous avons implémentée est son bon compromis entre une solution acceptable et le temps de calcul. En effet, celui-ci ignore beaucoup d'états lorsqu'il ajoute les enfants à la frontière. Cela a pour effet de ne pas garantir une solution optimale mais au moins le résultat peut arriver rapidement. Le point faible de notre algorithme est par contre qu'il ne garanti justement pas une solution idéale. Cela dépend des applications mais il se peut que ce point soit inacceptable.

Recherche locale

En utilisant la méthode du recuit simulé, nous pouvons nous échapper d'un minimum local ce qui peut être avantageux dans la recherche d'une bonne solution. Un des points faibles de notre implémentation est que notre solution de départ donnée par un algorithme vorace est très naïve ce qui peut donner du fil à retordre au recuit simulé pour converger. Finalement, les variables de température et de temps du recuit simulé ne sont pas tout à fait optimisées au problème.

Compétition

Utiliser la recherche en arbre pour la compétition.