Aide à jouer au Sudoku

Ren Shen

Alexandre Fernandes-Bartolomeu

2 mai 2017

Présenté à Pr. Roger Nkambou dans le cours INF4230 (Hiver 2017)

**1 Introduction**

Notre projet consiste à résoudre des jeux de Sudoku, tout en fournissant l’aide au joueur pour le guider dans ces choix de jeu. Ainsi, si nous avons un Sudoku au début avec aucune case de jouée sauf celles qui sont fournies au début, il sera possible d’afficher la solution, mais aussi de dire au joueur, par exemple, quand il se trompe et quel coup il devrait effectuer à la place pour réussir la grille. Notre Sudoku consiste en une grille de 9x9 dont le but est de remplir la grille de chiffres de 0 à 9 de façon à ce que :

1. Toutes les lignes possèdent tous les chiffres
2. Toutes les colonnes possèdent tous les chiffres
3. Toutes les carrés 3x3 dans la grille possèdent tous les chiffres

**2 Problématique**

Le problème que nous tentons de résoudre est d’implémenter une intelligence artificielle qui sera en mesure de compléter la grille, mais aussi de fournir et d’expliquer les coups que devrait prendre l’utilisateur en cas d’erreur. Ainsi, si l’utilisateur entre le chiffre 4 dans la première ligne, il faudra pouvoir lui dire que le chiffre ne peut aller à cet endroit et expliquer pourquoi, soit en montrant le fil d’exécution, soit en montrant le chiffre qui empêche ce chiffre d’être placé à cet endroit à ce moment. Dans le cas de notre projet, l’environnement est entièrement observable, car l’agent a accès à toute l’environnement à chaque instant. L’environnement est aussi déterministe, car l’état suivant, donc le prochain coup au Sudoku va dépendre de la grille courante. L’environnement est épisodique, car il s’agit d’observer la grille et de n’effectuer qu’une seule action par après. L’environnement est dynamique. Comme il s’agit d’un joueur qui joue au Sudoku et que l’intelligence artificielle ne fait qu’aider le joueur, il doit s’adapter aux coups que le joueur a effectués. L’environnement est discret, car il existe un nombre limité et distincts de données sensoriels et d’actions. En fait, le coup joué par l’utilisateur et l’état de la grille est tout ce dont à de besoin l’IA pour pouvoir décider de sa propre action. Finalement, il s’agit d’un environnement à agent unique, ou seule notre IA se déplace dans cet environnement.

**3 Technique d’IA retenue**

Le premier algorithme possible pour résoudre le problème de résolution du Sudoku est celui du *Backtracking Search*. Dans le cas du Sudoku, nous partons d’une case qui consiste au nœud parent, puis on passe au premier nœud fils qui satisfait les contraintes, énumérées dans l’introduction. Si tous les nœuds fils satisfont les contraintes, alors l’algorithme a trouvé la solution. Si on arrive au dernier fils du premier nœud parent et que celui-ci ne satisfait pas les contraintes, alors il n’existe pas de solution.

Le deuxième algorithme possible est l’algorithme stochastique. Avec cet algorithme il faut tout d’abord remplir la grille, puis regarder tous les doublons qu’il y a pour chaque lignes et colonnes. Une fois que l’on obtient les doublons pour chacun, il faut regarder quel carré 3x3 possède le plus de doublons, en combinant les doublons des lignes et des colonnes pour le carré 3x3. Il faut ensuite prendre ce bloc et le modifier de façon aléatoire jusqu’à obtenir moins de doublons qu’avant pour la grille. Lorsque cela est réussi, il faut recalculer les doublons pour choisir quelle case 3x3 modifier, jusqu’à arriver à zéro doublons pour toute la grille, ce qui voudra dire que la solution a été trouvée.

http://igm.univ-mlv.fr/~dr/XPOSE2013/sudoku/index.html