

Modélisation distribuée d'un jeu stratégique : Cas d'une variante du jeu de dames

CHUPIN Pierre-Henri
SOLOMON Maria

Abstract

Ce projet a pour but de créer une modélisation distribuée et de l'appliquer à un cas concret ici une variante du jeu de dames. Cela permettra de créer des stratégies pour des jeux et de comprendre mieux comment fonctionne la modélisation distribuée.

Mots-clés: Modélisation distribuée, stratégie, recherche de risque

1 Introduction

Ce projet s'inscrit dans le cadre du second semestre de Master 1 informatique de Lyon et dans l'optique de poursuivre en Master IA de Lyon. Pour ce projet nous avons pu compter sur notre encadrant Aknine Samir appartenant à l'équipe Systèmes Multi-Agents de Lyon. Depuis longtemps beaucoup d'informaticiens veulent résoudre des jeux comme les échecs en créant des IA capable de faire cela. Résoudre ce genre de problèmes demande beaucoup de travail car il faut dans un premier temps créer informatiquement le jeu que l'on veut créer et mettre en place toutes ses règles. Une fois tout cela fait le vrai travail commence, le codage des stratégies que l'IA utilisera pour jouer au jeu. Nous avons décidé de faire la modélisation distribuée d'un jeu ancestrale, le jeu de dames.

2 Etat de l'art

2.1 Modélisation Distribuée

3 Outils utilisés

3.1 Le Jeu de dames

La modélisation réalisée se base sur le jeu de dames classique mais avec quelques changements de règles qui permettent d'avoir des comportements différents de ce qu'on peut voir en général. voici la liste des différentes règles et système utilisé :

- Un plateau carré de 10 cases sur 10
- 20 pions blancs pour un joueur et 20 pions noirs pour un autre joueur

- chaque joueur jouera à tour de rôle et il pourra jouer 1 à N (nombre choisi avant de lancer une partie) pièces en même temps ce qui permet de voir comment vont réagir plusieurs pièces ensemble
- Création classique de la dames qui permet d'avoir plusieurs types de pièce à gérer
- Une dame peut se déplacer sur 3 cases de diagonale maximum nous avons voulu changer avec les règles habituels
- La prise est obligatoire comme dans les règles classique des dames.
- La prise multiple est possible aussi comme dans les dames classiques
- Une partie est gagné si on prend toutes les pièces adverse ou si l'adversaire ne peut pas se déplacer quand c'est son tour de jeu.
- Si il y a 30 coups sans prise la partie est nul

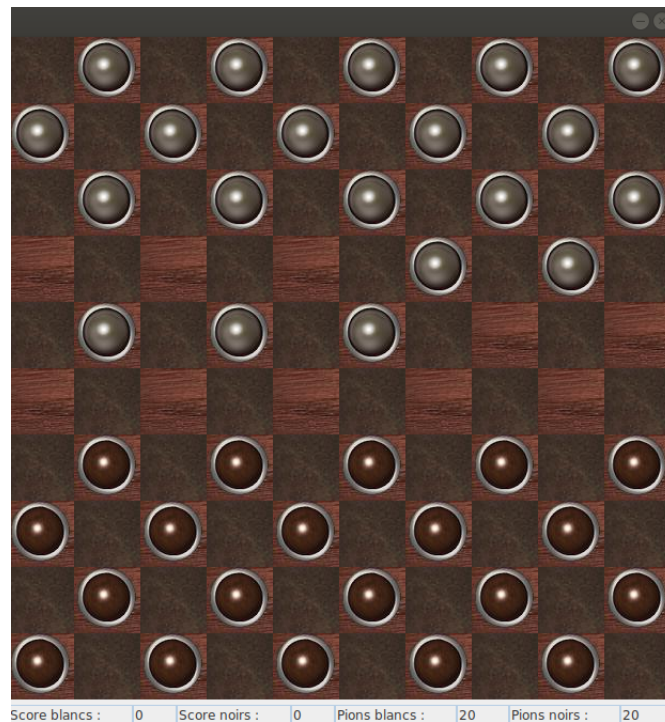
4 Travail réalisé

4.1 Modélisation du jeu de dame

Afin de commencer notre travail sur la modélisation distribuée nous avons dû bien sûr faire la modélisation du jeu de dame. Le fait de donner un aspect graphique au jeu (on voit le plateau et les pièces se déplacer) était important afin de mieux voir ce qu'il se passe pendant le déroulement du programme et donner un côté graphique à notre travail. Bien sûr on aurait pu le faire tourner sans affichage pour juste avoir le résultat de la simulation. Il a donc fallu partir à la recherche d'image pouvant être intégrée au projet qui permettrait de rendre cela possible. il nous fallait alors

- Une image de plateau carré de 10 cases sur 10
- Une image de pion blanc et noir
- Une image de dame blanche et noir

Une fois cela fait il nous suffit de mettre en place le plateau avec nos différentes images [?]. Voici donc à quoi ressemble le plateau et les pièces après le premier coup des blancs.



afin de mieux pouvoir voir ce qu'il se passe il a fallu mettre en place la possibilité de mettre en pause le jeu[?] pour s'arrêter et comprendre au mieux comment le programme fonctionne, nous verrons par la suite des exemples précis de comportement.

4.2 Stratégie Naive

Avant de pouvoir commencer à aller loin dans la modélisation distribuée nous avons dû faire une stratégie naive qui consiste simplement à faire jouer l'IA le plus simplement possible sans aucune réflexion de sa part. Nous sommes partis sur le principe que l'IA jouera le maximum de pions possible pour chaque coup. Dans un premier temps elle regardera les coups obligatoires à faire comme la prise et jouera toutes les prises, si il lui reste des mouvements possible elle devra alors regarder chaque pièce non déplacée si elle peut les déplacer. Si oui alors elle la joue qu'importe si le mouvement est mauvais ou non jusqu'à ne plus avoir de mouvement possible à jouer. cette stratégie naive nous servira de base pour créer une stratégie plus performante par la suite.

4.3 Amélioration de la prise

4.4 Amélioration du déplacement

Il y avait comme soucis majeur le choix des pièces à déplacer une fois les coups obligatoires joués. Nous avons donc dû réfléchir à comment faire pour éviter les mouvements mauvais (qui nous font perdre des pièces juste après). Pour ce faire nous avons

cherché des algorithmes connus qui sont utilisés pour ce genre de problème. C'est le cas de l'algorithme min-max qui nous a inspiré. Le but est de tester à chaque fois qu'on prévoit de déplacer une pièce si cela est risqué pour elle ou non. Pour ce faire nous avons eu besoin de créer une fonction récursive qui teste si la pièce que l'on veut jouer sera en danger si on l'a jouée. Cela fait partie de la première étape de la fonction car le but de l'algorithme min-max est de fabriquer un arbre de toutes les possibilités possibles de mouvement et de descendre le plus possible en profondeur. Le problème est tant que l'arbre devient extrêmement grand très rapidement. Exemple si on peut déplacer 10 pièces différentes et chacune à deux positions différentes on aura un arbre de degré 20 sur le nœud racine mais cela ne s'arrête pas là car si on veut descendre d'un cran et voir le déplacement que l'adversaire pourrait faire en réponse alors on a pour chaque nœud tous les mouvements possibles de l'adversaire et rappelons que nous donnons la possibilité de déplacer au choix 1 à N pièces par coups ce qui augmente le nombre de déplacements adversaires possibles et donc augmente la taille de l'arbre. C'est pour cette raison que nous avons voulu limiter notre parcours en profondeur de l'arbre et la façon dont nous cherchons dans l'arbre. Au lieu de simuler tout l'arbre puis de chercher dedans nous le simulons petit à petit et nous arrêtons dès que l'on a trouvé un mouvement satisfaisant. Pour ce faire nous prenons les pièces une à une et quand elle peut se déplacer alors nous testons si après avoir fait ce déplacement elle risque de se faire prendre par l'adversaire et si c'est le cas alors nous la jouons pas. Au contraire si elle peut se déplacer sans risque alors elle est jouée et nous arrêtons de vérifier dès que nous avons joué le nombre maximum de pièces possible dans notre coup ce qui fait que nous créons petit à petit l'arbre des coups possibles afin de limiter la taille de celui-ci. Bien sûr cela implique des limites car certaines fois un coup sans risque par rapport à un autre est plus efficace.

5 Conclusion