

---

# Apprentissage par renforcement dans une approche Monte-Carlo du jeu de Go

**Guillaume Chaslot\*** — **Rémi Coulom\*\*** — **Bruno Bouzy\*\*\*** — **Philippe Preux\*\***

\* *École Centrale de Lille*

*chaslot.guillaume@ec-lille.fr;*

\*\* *Groupe Apprentissage Automatique (GRAPPA)*

*Université Charles de Gaulle – Lille 3*

*{Remi.Coulom, Philippe.Preux}@univ-lille3.fr*

\*\*\* *Centre de Recherche en Informatique de Paris 5 (CRIP5)*

*Université René Descartes – Paris 5*

*bouzy@math-info.univ-paris5.fr*

---

*RÉSUMÉ. Le cadre des travaux présentés dans cet article est celui de la l'amélioration d'une fonction d'évaluation de type Monte Carlo pour le jeu de Go. Une évaluation Monte Carlo est la moyenne des résultats terminaux d'un grand nombre de parties aléatoires simulées à partir de la position courante. La difficulté est de déterminer une distribution de probabilité pour la sélection aléatoire des coups, de manière à obtenir la meilleure fonction d'évaluation possible. Cet exposé présentera différentes techniques d'apprentissage de cette distribution, et les résultats expérimentaux obtenus.*

*ABSTRACT. The objective of the research presented in this paper is to learn an evaluation function for the game of Go. More precisely, this evaluation function is of the Monte-Carlo type, where the value is computed by averaging terminal results of a large number of random games simulated from the current position. The difficulty is in determining a probability distribution for the random selection of moves, in order to obtain the best possible evaluation. We will present learning methods to learn such a probability distribution, as well as experimental results.*

*MOTS-CLÉS : apprentissage par renforcement, Monte-Carlo, jeu de Go*

*KEYWORDS: reinforcement learning, Monte-Carlo, game of Go*

---

## 1. Introduction

Le jeu de Go (<http://jeudego.org> pour une introduction) est un jeu de réflexion qui pose des problèmes difficiles à la recherche en intelligence artificielle [BOU 01]. Contrairement à de nombreux autres jeux classiques, comme les échecs, les meilleurs programmes jouant au Go ont un niveau très faible. Même si les programmes sont capables de battre les débutants, ils se font tous battre par des humains amateurs.

Une des difficultés fondamentales dans l'élaboration d'un programme fort est la construction d'une fonction d'évaluation des positions. Dans des jeux comme les échecs ou les dames, il est simple de construire une heuristique d'évaluation. On peut par exemple compter le nombre de pions (avec, aux échecs, fou et cavalier valant 3 pions, tour valant 5 pions et Dame valant 9 pions, par exemple), ce qui constitue une très bonne base pour estimer la valeur d'une position. Il n'existe pas de telles heuristiques efficaces dans le jeu de Go.

Comme la construction d'heuristiques statiques simples est très difficile pour le Go, des approches dynamiques de type Monte-Carlo sont intéressantes [BOU 04, SHE 04]. Le principe consiste à évaluer une position avec la moyenne des valeurs terminales d'un grand nombre de parties simulées à partir de cette position. Dans chaque partie, les coups sont sélectionnés aléatoirement. La valeur terminale d'une partie peut être le gain (1 pour gagné, -1 pour perdu, par exemple), ou le score (la différence de points entre les deux joueurs).

La qualité d'une telle évaluation de type Monte Carlo dépend grandement de la distribution de probabilité utilisée pour le choix des coups. Cet exposé présente des méthodes pour l'optimisation automatique d'une telle distribution de probabilité, ainsi que les résultats expérimentaux obtenus.

## 2. Bibliographie

- [BOU 01] BOUZY B., CAZENAVE T., « Computer Go : an AI-Oriented Survey », *Artificial Intelligence Journal*, vol. 132, 2001, p. 39–103.
- [BOU 04] BOUZY B., « Associating domain-dependent knowledge and Monte-Carlo approaches within a go program », *Information Sciences*, , 2004.
- [SHE 04] SHEPPARD B., « Efficient control of Selective Simulations », *International Computer Games Association Journal*, vol. 27, n<sup>o</sup> 2, 2004, p. 67–79.