

Algorithmes approchés

Arpad Rimmel

SUPELEC

4 janvier 2013

Table des matières

- 1 Introduction
- 2 Travaux précédents
- 3 Travaux actuels
- 4 Perspectives

Introduction

Catégories de problèmes étudiés :

- Problèmes ayant une complexité trop importante pour être résolue en un temps raisonnable.
Exemple : exploration d'arbre avec un grand nombre de nœuds.
- Problèmes où il est impossible de trouver une solution optimale.
Exemple : optimisation d'une fonction continue sans propriété.

Utilisation d'algorithmes donnant la meilleur solution possible en un temps donné.

Table des matières

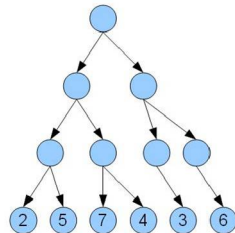
- 1 Introduction
- 2 Travaux précédents**
- 3 Travaux actuels
- 4 Perspectives

Exploration d'arbre

Problème : prendre des décisions dans un environnement discret, observable, avec horizon fini et avec récompenses.

Exemple :

- multiplication de matrices
- jeu de Go
- samegame
- POMDP
- ...



Bandit Based Monte Carlo Tree Search

- Basé sur la formule du bandit
- Évaluation par simulation Monte Carlo
- Performant pour les applications où la première décision est importante

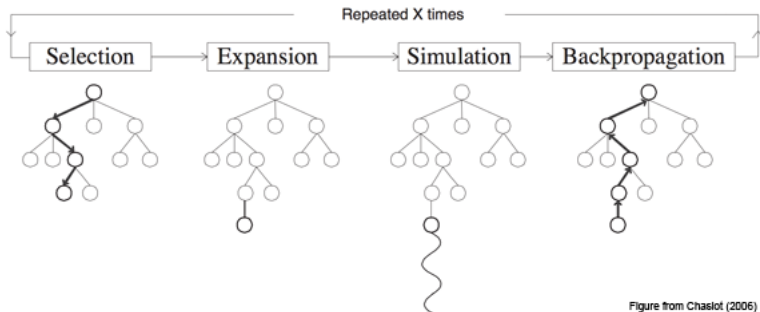
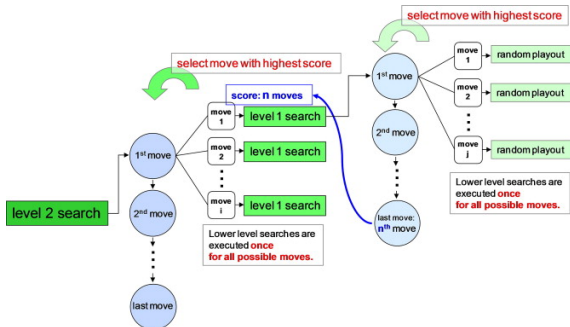


Figure from Chaslot (2006)

Nested Monte Carlo Tree Search

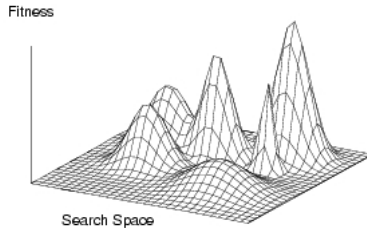
- Basé sur une récursion d'évaluations
- Évaluation par simulation Monte Carlo
- Performant pour les applications où toutes les décisions sont importantes



Optimisation

Problème :

- Trouver le maximum d'une fonction "boite noire".
- on peut obtenir la valeur en un point donné.
- Espace de recherche trop grand pour être exploré en entier.



Algorithmes Evolutionnaires

- Basé sur l'évolution d'une population d'individus
- Chaque individu correspond à un point
- Le score de chaque individu correspond à la valeur de la fonction en ce point
- De nouveaux individus sont générés à partir des scores précédents

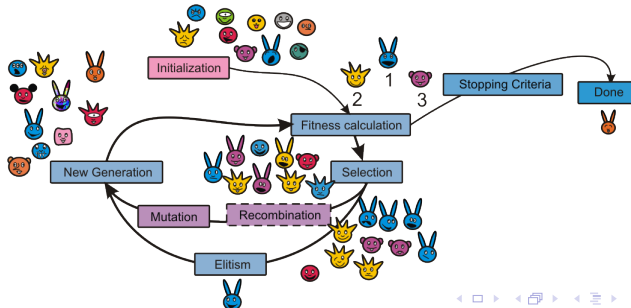


Table des matières

- 1 Introduction
- 2 Travaux précédents
- 3 Travaux actuels**
- 4 Perspectives

Carrés latins

But :

- Placer n points dans une hypercube de taille n par n .
- todo : pour une dimension donnée, chaque point doit avoir une valeur différente.
- todo : trouver la solution où la distance minimale entre 2 points est maximale.

Applications :

- Planification d'expériences

X			
	X		
			X
		X	

		X	
X			
			X
	X		

Etat de l'art

- En dimension 2, pour les normes L_1 et L_{inf} , des algorithmes donnant la solution optimale en un temps linéaire ont été fournis.
- Pour la norme L_2 , uniquement des solutions approchées.
- En dimension supérieure à 2, uniquement des solutions approchées.

Travaux préliminaires

Pour le moment 2 algorithmes testés :

- Nested Monte Carlo Tree Search
Problème : pas vraiment une structure d'arbre.
- Algorithme génétique
Résultat prometteurs.

Table des matières

- 1 Introduction
- 2 Travaux précédents
- 3 Travaux actuels
- 4 Perspectives**

Perspectives

todo

- trouver des chemins dans des graphes
- trouver des colorations pondérées