University of Basel

ASAP-UCT:TO DO

Bachelor thesis

Natural Science Faculty of the University of Basel
Department of Mathematics and Computer Science
Artificial Intelligence
http://ai.cs.unibas.ch

Examiner: Prof. Dr. Malte Helmert
Supervisor: Dr. Thomas Keller

Tim Steindel
tim.steindel@stud.unibas.ch
2014-050-389

02/11/2017

Acknowledgemnts

Calculations were performed at sciCORE (http://scicore.unibas.ch/) scientific computing core facility at University of Basel.

Abstract

Missing: ABstract schreiben

Table of Contents

Acknowledgemnts **ii**

Abstract **iii**

1 Introduction **1**

2 Background **2**

2.1 Markov Decision Process 2

2.2 tree 2

2.3 Upper Confidence Bounds to Tree (UCT) 3

2.3.1 Selection 4

2.3.2 Expansion 4

2.3.3 Simulation 4

2.3.4 Back-propagation 4

3 ASAP-UCT **5**

4 Evaluation **7**

5 Conclusion **8**

Bibliography **9**

Appendix A Appendix **10**

Declaration on Scientific Integrity **11**

1 Introduction

In traditional Monte Carlo Tree Search(MCTS) we have a flat tree , which lead to a scaling problem, if we have a large problem. An often used method to reduce the number of domain feature, and thus reduces computation is trough domain abstraction.

In our case we compute equivalence classes of states,but furthermore we also compute the equivalence class of state-action pairs.

2

Background

2.1 Markov Decision Process

Many planing problem can be modeled by a Markov Decision Process (MDP), which is defined as a 6-Tuple $< S, A, T, R, s_0, H >$ with:

- S: a finite set of states
- A: a finite set of action
- T: the probability to transit from $s \in S$ to $s' \in S$
- $R : S$ x A x $S \to \mathbb{R}$ is the reward function
- s_0 the initial state
- H $\in \mathbb{N}$ is the finite horizon

Where a new state is only dependent of his previous state and an action.

2.2 tree

The tree is defined as a 6-tuple $< D, C, V_c, V_d, n_0, L_c >$ where

- D: finite set of Decision Node
- C: finite set of Chance Node
- V_c: are the edges which connect a decision node with a chance node
- V_d $\in$[0,1]: are the edges which connect a chance node with decision node
- $n_0 \in D$ is the root node
- L_c are the actions

Definition *Decision Node* is a 3-Tuple $< s, \hat{V}, N >$ where

- s is a state

- $\hat{V}$ is the state-estimate
- N is the number of visit of this node

Definition *Chance Node* is a 4-Tuple $< s, a, \hat{Q}, N >$ where

- s is a state
- a is a action
- $\hat{V}$ is the state-estimate
- N is the number of visit of this node

The tree has some basic rules :

- the root node is a decision node
- each type of nodes has only children of the other type
- leaves of this tree are chance nodes

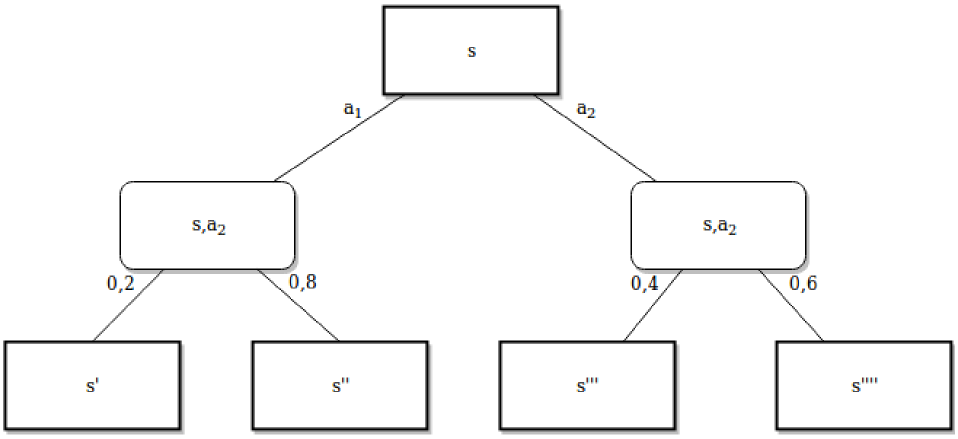

Figure 2.1: after calculating the equivalence classes , each color simulate a different Equivalence class

2.3 Upper Confidence Bounds to Tree (UCT)

Because it is not optimal to traverse the whole tree we need to evaluate action that are promising and expand in this direction. For this we use a tree made with two different types of nodes: *Decision Nodes* and *Chance Nodes*.

The UCT algorithm is structured in 4 phases:
Selection, **Expansion**, **Simulation** and **Back-propagation**

2.3.1 Selection

In this phase , there are two possibility: **Exploitation** use promising candidate or **Exploration** where the candidate is not well examined .Beginning in the root node the algorithm calculate f the *Upper Confidence Bound (UCB)* or each children and goes to the child where $UCB = argmax_{i \in I}(v_i + C\sqrt{\frac{ln\ n_p}{n_i}})$
where I is the set of children of p, v_i is the average reward for i and is the exploitation value, n_p and respectively n_i the number of visit for p respectively i and C a constant to determine how curiosity the algorithm is, with a small C the tree becomes deeper, a large C makes the tree wider.

2.3.2 Expansion

If the Selection comes to the point , where the current nodes has less expanded children than a threshold T new child or children will be here expanded.

2.3.3 Simulation

In this phase the reward of the new node is estimated.

2.3.4 Back-propagation

After the Simulation-phase the tree is outdated. Therefore beginning in the newest node the algorithm traverse the tree to the root node and update all the nodes en route.

3
ASAP-UCT

The following Chapter shows the implementation of ASAP-UCT.
For calculation of the equivalence classes we need 2 definition:
Definition 1: Decision Node Abstraction

Suppose we have two decision node , they are in the same equivalence class, if for each children c of the first decision node, there is also a child c' in the second node with the same equivalence class and same probability.
Definition 2: Chance Node Abstraction

A Chance Node n is in the same equivalence with a second chance node n_2, if for each successor state there is a decision node with the same equivalence class and the same probability or they are on the same level and are both leaf nodes.

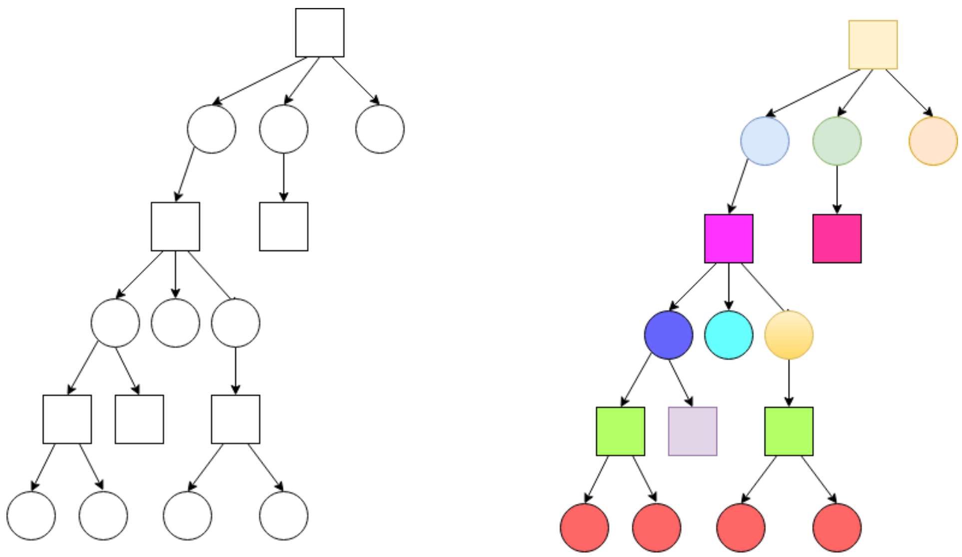

Figure 3.1: after calculating the equivalence classes , each color simulate a different Equivalence class

After a given time interval τ we recursively generate the equivalence classes beginning from the leaves of the UCT-tree and working our way up. With the finished equivalence classes we can now use the Q-value of the equivalence classes in our original tree for the back-propagation. With this projection we have the advantage that we do not have a second abstract tree and therefore have no outdated tree.
But because we do not abstract between level, we still have the same depth.

4

Evaluation

5

Conclusion

Bibliography

[1] Turing, A. M. Computing machinery and intelligence. *Mind*, 59(236):433–460 (1950).

[2] Turing, A. M. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London mathematical society*, 42(2):230–265 (1936).

A

Appendix

Declaration on Scientific Integrity
Erklärung zur wissenschaftlichen Redlichkeit

includes Declaration on Plagiarism and Fraud
beinhaltet Erklärung zu Plagiat und Betrug

Author — Autor
Tim Steindel

Matriculation number — Matrikelnummer
2014-050-389

Title of work — Titel der Arbeit
ASAP-UCT:TO DO

Type of work — Typ der Arbeit
Bachelor thesis

Declaration — Erklärung
I hereby declare that this submission is my own work and that I have fully acknowledged the assistance received in completing this work and that it contains no material that has not been formally acknowledged. I have mentioned all source materials used and have cited these in accordance with recognised scientific rules.

Hiermit erkläre ich, dass mir bei der Abfassung dieser Arbeit nur die darin angegebene Hilfe zuteil wurde und dass ich sie nur mit den in der Arbeit angegebenen Hilfsmitteln verfasst habe. Ich habe sämtliche verwendeten Quellen erwähnt und gemäss anerkannten wissenschaftlichen Regeln zitiert.

Basel, 02/11/2017

Signature — Unterschrift