When to Commit to an Action in Online Planning

source article.

Plan

- Introduction.
- Problem setting.
- Proposed method.
- Committing VS not committing .
- Summary.

Introduction

- When a planner commits to an action, it re-roots it's search tree at the node that was a result of that action.
- A time presser results in real time search, when the planner have to commit to a new action before the previously one ended. This action can be a no-operation action, that keeps the state unchanged.
- Time presser can be beneficial to commit early to actions, so that more look ahead search are performed on upcoming states.
- The paper proposes a method for making commitment decision.

Problem setting

- The system must be controlled all the time (actions must be executing at any time).
- The planner can commit to actions earlier in order to re-root the tree at a deeper node →focus the search later on the future.
- <u>Assumptions:</u>

1. Actions are serials.
2. The world is observable and deterministic.

- <u>Objective of the system:</u>

Achieve the goal as soon as possible.

Proposed method

- A meta-reasoning scheme for action commitment called Flexible Action Commitment Search (FACS).
- The paper used additional assumption to estimate the utility of the committing or not committing to an action and to simplify the analysis such as:

1. Each node has exactly tow children.
2. The time required to fully execute an action is identical for all actions.
3. The order of decision is a fixed search tree.
4. No re-planning after action commitment.
5. The search my re-started if necessary.

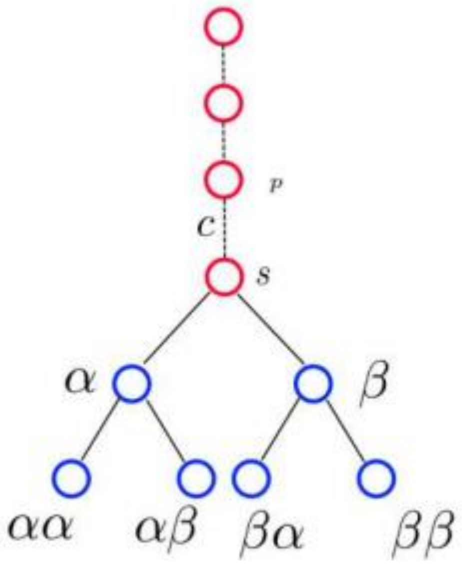
Committing VS not committing

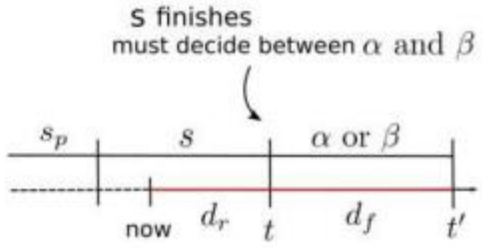
- During the search, after each expansion or periodically after a set of expansions a meta reasoning process decides whither:

- Commit to the current top-level action **now and** re-root the search from that node, **or**
- postpone the commitment and continue the search.

Committing VS not committing

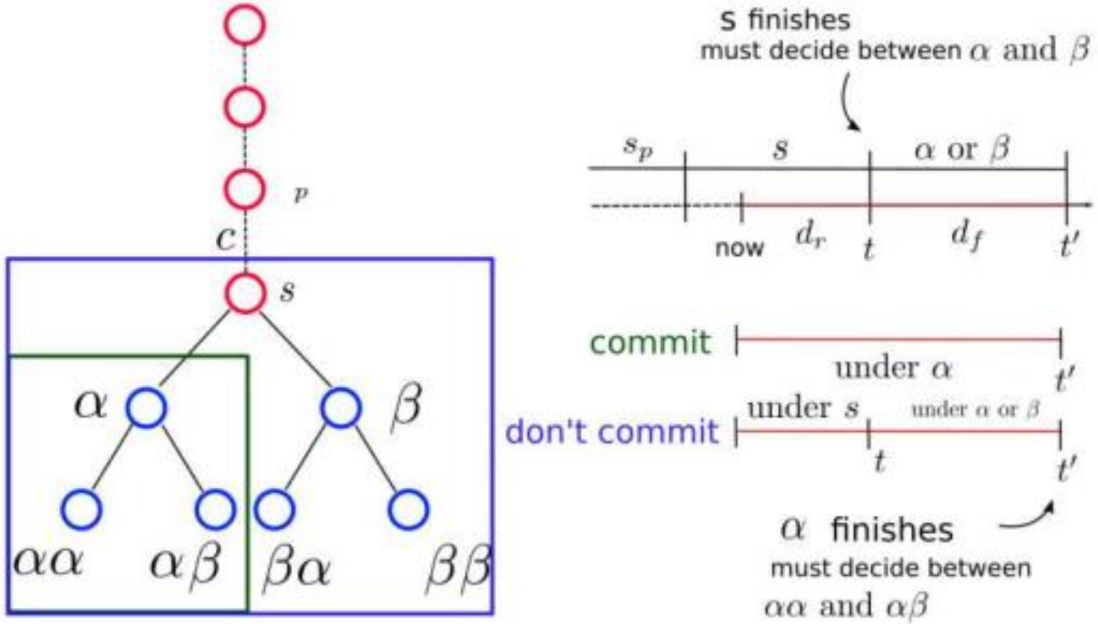
- Current search tree rooted at node s.
- Available search time denoted by d consist of d_r and d_f
- d_r: is the remaining time induced by previous commitments
- d_f :The time required to execute a full action.





Committing VS not committing

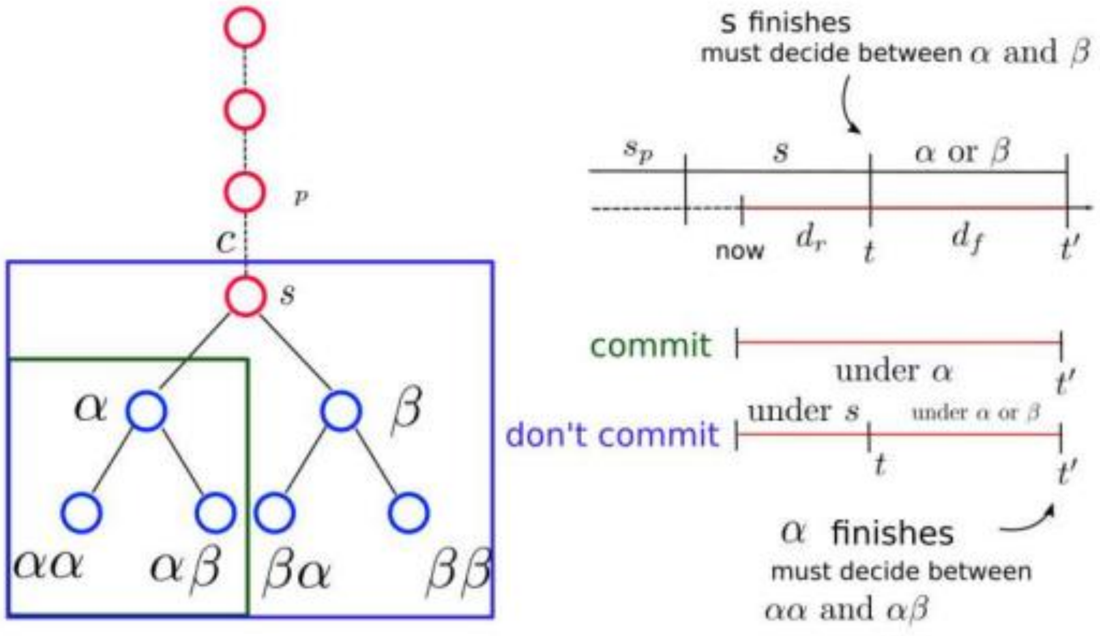
- By committing to α, all the available search time is used to search under children of α (αα and αβ).each one receives a search budget of d = (dr+df)/ 2.
- The utility of committing to α is the expectation of the minimum ˆf-value of αα and αβ, after searching d time units under each of them: Ucommit = E [min(Xd αα, Xd αβ)]



Committing VS not committing

- If choosing to not commit yet. Half of dr (dr/2) is used to search under each child of the root (s).
- The rest of the time df is used to search under whichever child of the root is judged most promising at that time.
- The search duration under each grand child is

d' = ((dr/2) +df) / 2.



Summary

- Committing means re-rooting the search tree.
- Not committing means to not re-root the search tree.
- Algorithms committing to one action at a time and re-root the search tree at every step cannot benefit from gaining future searching time.
- The method proposed in this paper focuses on real time search to get more time to search ahead in the search tree.
- The evaluation of the method outperform previously preposed fixed strategies such as LSS-LRTA*(all), LSS-LRTA*(one) and Dynamic f.