When to Commit to an Action in Online Planning

Tianyi Gu\textsuperscript{1} and Wheeler Rum\textsuperscript{1} and Shahaf Shperberg\textsuperscript{2} and Eyal Shlomo Shimony\textsuperscript{2} and Erez Karpas\textsuperscript{3}

\textsuperscript{1} University of New Hampshire
\textsuperscript{2} Ben-Gurion University of the Negev
\textsuperscript{3} Technion Israel Institute of Technology
Classical Planning Environments:

- single agent
- discrete state, discrete action
- complete observability
- deterministic state transition
- online planning: interleaving planning and execution
An example: highway navigation

agent performs search for a bounded time
An example: highway navigation

agent performs search for a bounded time
An example: highway navigation

agent commits to best action and executes
An example: highway navigation

agent commits to best action and executes
An example: highway navigation

online planning: interleaving search and action execution
“receding horizon control”
The Meta-level Problem: Commit or Not Commit

For each node along the best prefix path: should we commit?
For each node along the best prefix path: should we commit?

fixed strategies:
- always commit one (Korf 1990)
- always commit all (Koenig & Sun 2008, Burns et al 2013)

Can we do better?
always commit all is too risky
The Meta-level Problem: Commit or Not Commit

Introduction

Online Planning
An Example
Action Commitment

FACS
Results
Conclusions

always commit all is too risky

always commit one is too conservative
ideal:

commit if an action in prefix is certainly the best to gain more planning time for next iteration
Flexible Action Commitment Search
Assumptions

- System can’t be uncontrolled, so force to commit if action queue is empty.
- Search tree structure (order of decisions is fixed).
- Deterministic system (no replanning required).

Only deal with commitment strategy.
we propose a principled way to make meta-level decision
we propose a principled way to make meta-level decision
we propose a principled way to make meta-level decision
belief of where \( \hat{f} \) will be after search:
belief of where $\hat{f}$ will be after search:

$$X_{\alpha\alpha}^d \sim \mathcal{N} \left( \hat{f}, \text{var}(dtg, d) \right)$$
$U_{\text{commit}} = \mathbb{E} \left[ \min(X_{\alpha\alpha}^d, X_{\alpha\beta}^d) \right]$

where $d = (d_r + d_f)/2$

$U_{\text{don't commit}} = P_{\text{choose } \alpha} \cdot U_\alpha + (1 - P_{\text{choose } \alpha}) \cdot U_\beta$

commit when $U_{t_{\text{commit}}} > U_{t_{\text{don't commit}}}$
Results
Left: tar pit area $\rightarrow$ high cost for reckless committing
- Right: corridor area $\rightarrow$ need long lookahead to observe the local minima
- Middle: empty area $\rightarrow$ gain lookahead, no harm to commit
map with only tar pit

commit-all perform badly
Results

map with only corridor

algorithms with small action queue perform badly
FACS consistently performs the best.
Summary

- FACS starts to explore a principled way of doing online action commitment
- FACS is better than fixed baseline strategies in synthetic grid pathfinding scenarios.
- Deliberation on how to allocate search effort can benefit online planning

More broadly:
- **Metareasoning** pays off when planning under time pressure!