When to Commit to an Action in Online Planning

Tianyi Gu[1] and Wheeler Ruml[1] and Shahaf Shperberg[2] and
Eyal Shlomo Shimony[2] and Erez Karpas[3]

[1] University of New Hampshire

[2] Ben-Gurion University of the Negev

[3] TECHNION Israel Institute of Technology

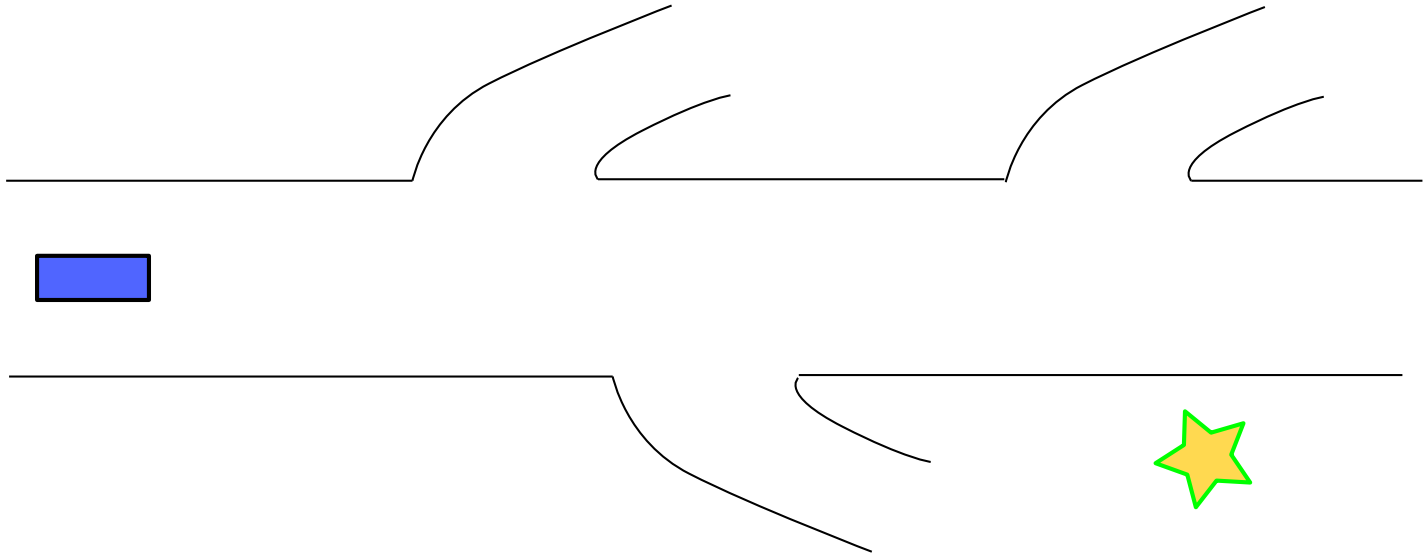
Problem Setting: Plan While Action Execution

Classical Planning Environments:

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

An Example of Online Planning Using Heuristic Search

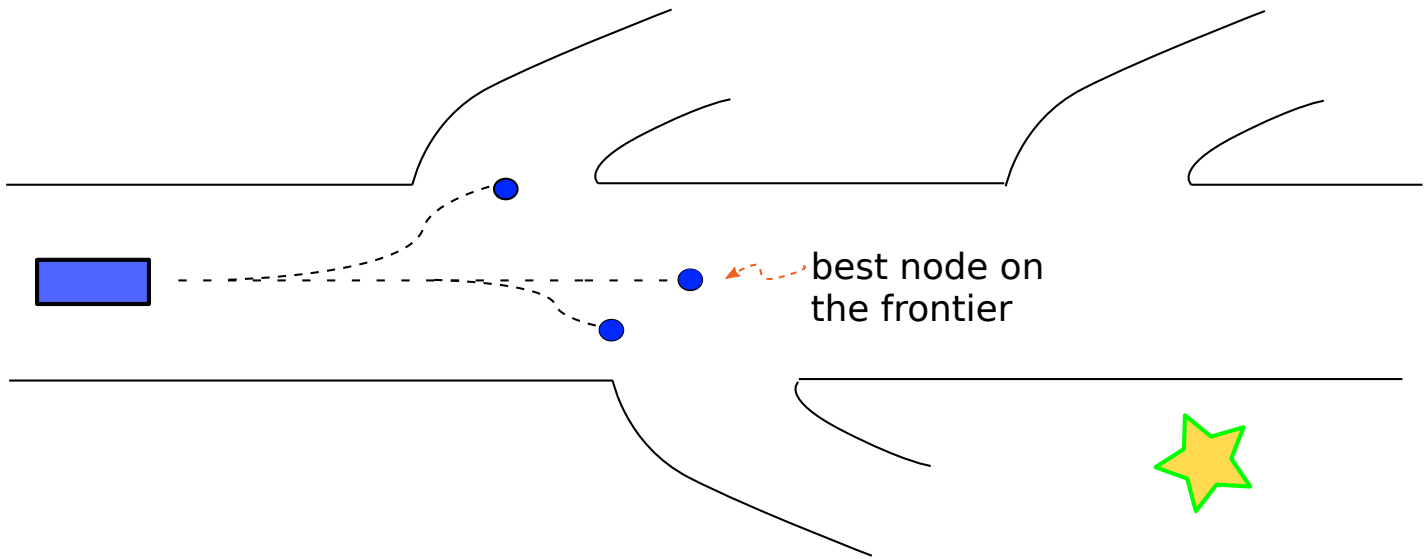
An example: highway navigation



agent performs search for a bounded time

An Example of Online Planning Using Heuristic Search

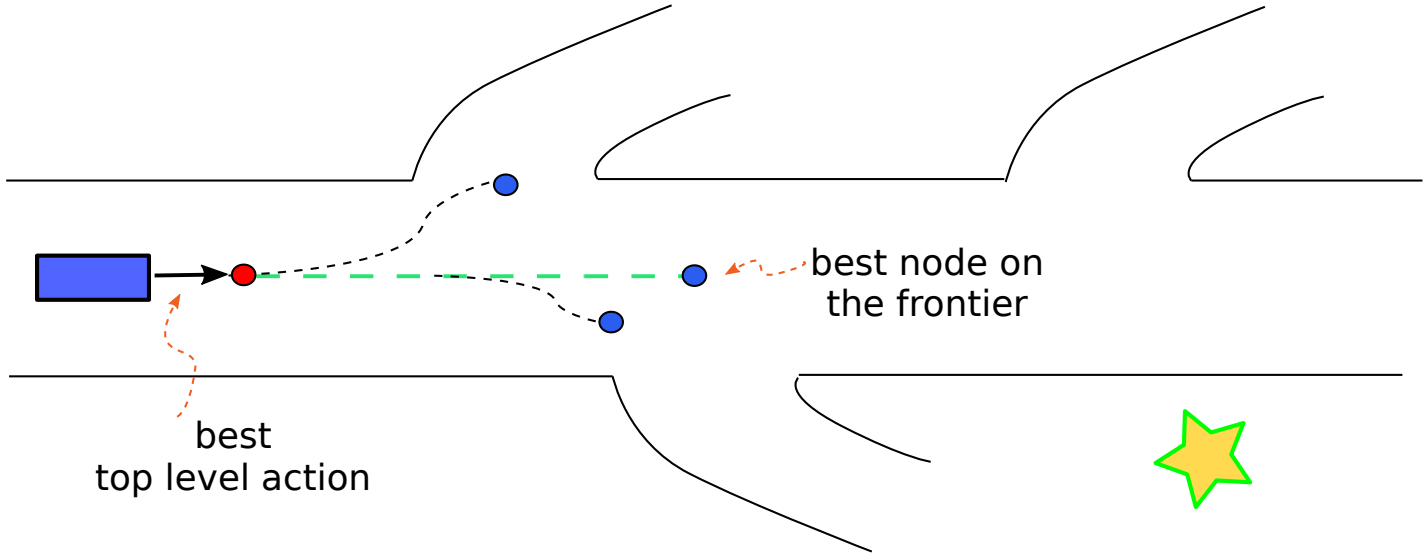
An example: highway navigation



agent performs search for a bounded time

An Example of Online Planning Using Heuristic Search

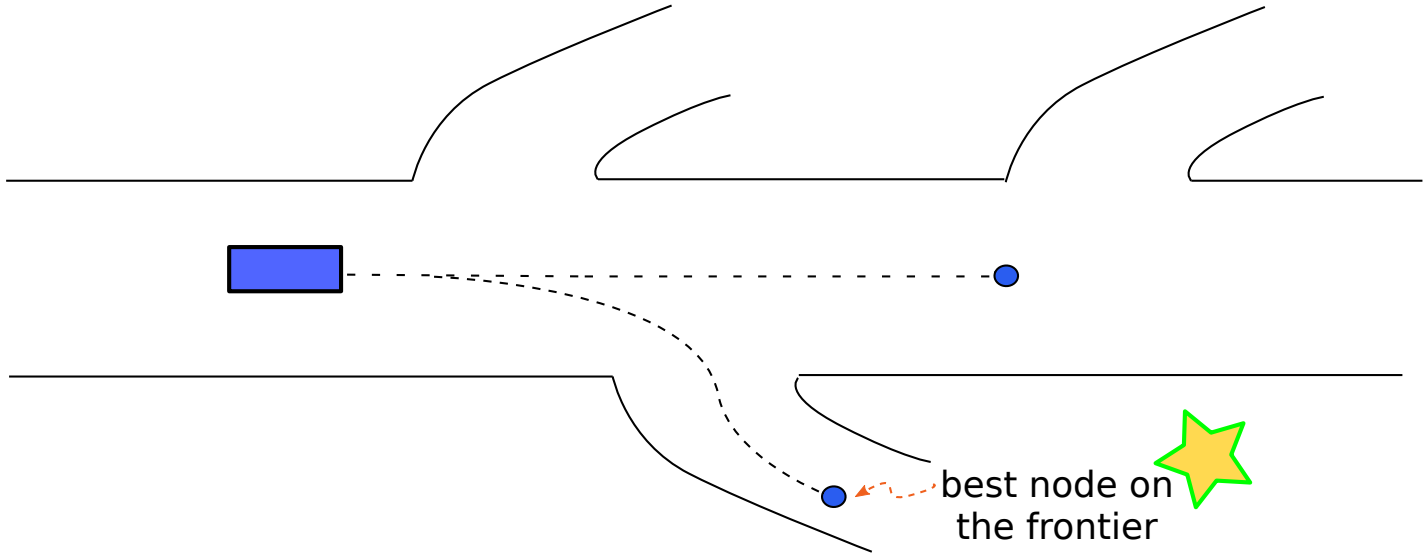
An example: highway navigation



agent commits to best action and executes

An Example of Online Planning Using Heuristic Search

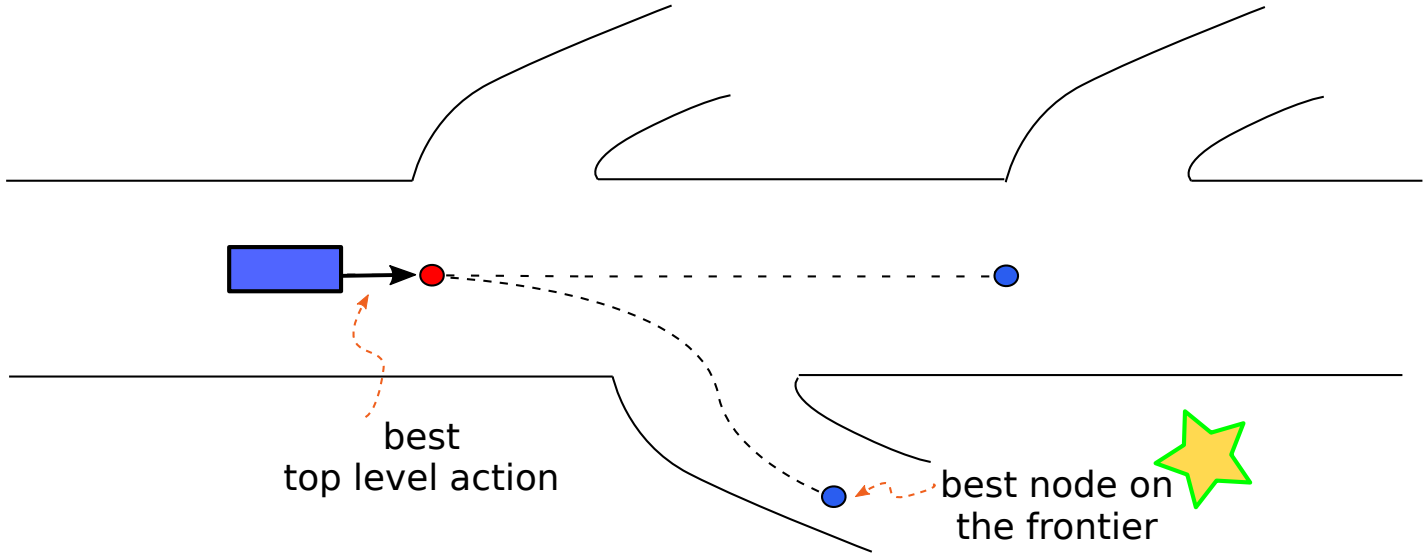
An example: highway navigation



agent commits to best action and executes

An Example of Online Planning Using Heuristic Search

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?

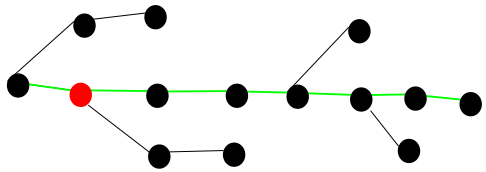
The Meta-level Problem: Commit or Not Commit

For each node along the best prefix path:

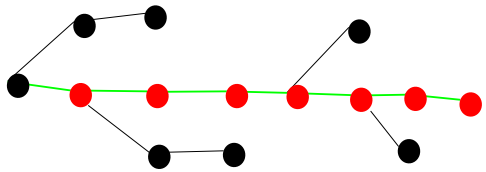
should we commit?

fixed strategies:

always commit one (Korf 1990)



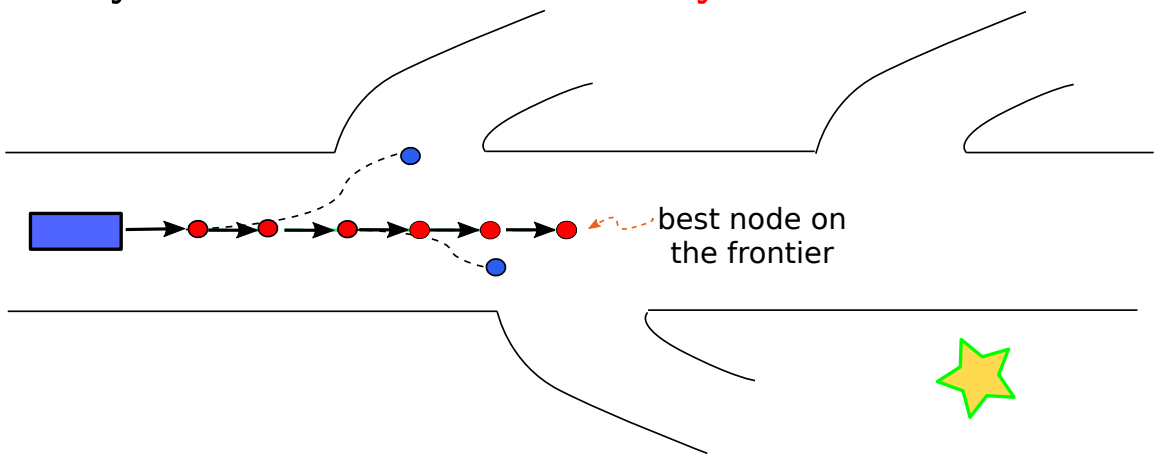
always commit all (Koenig&Sun 2008, Burns et al 2013)



Can we do better?

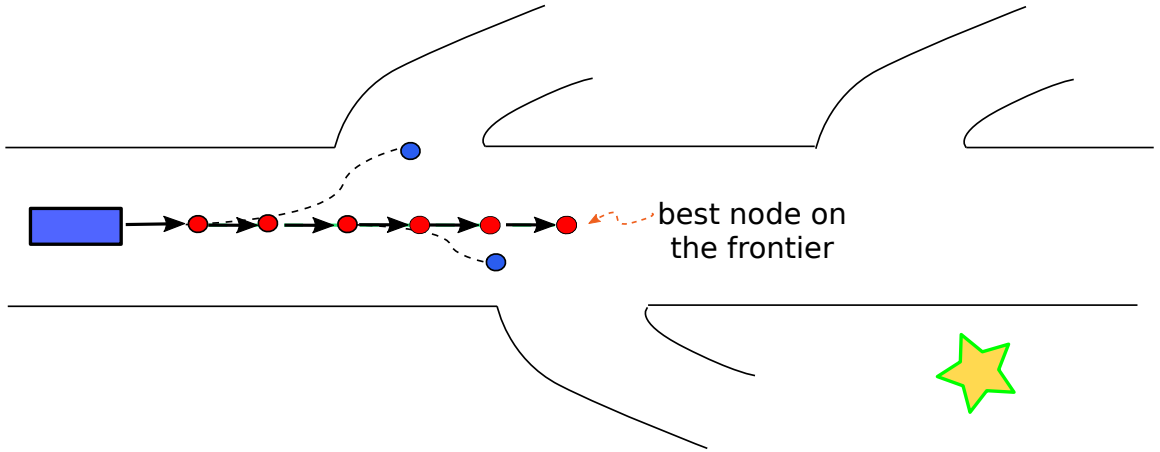
The Meta-level Problem: Commit or Not Commit

always commit all is too risky



The Meta-level Problem: Commit or Not Commit

always commit all is too risky



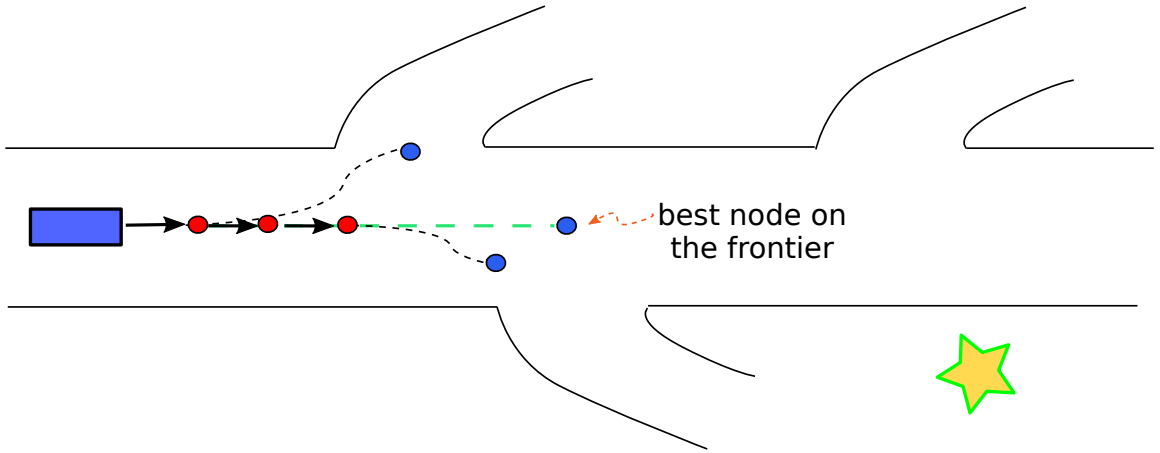
always commit one is too conservative

The Meta-level Problem: Commit or Not Commit

ideal:

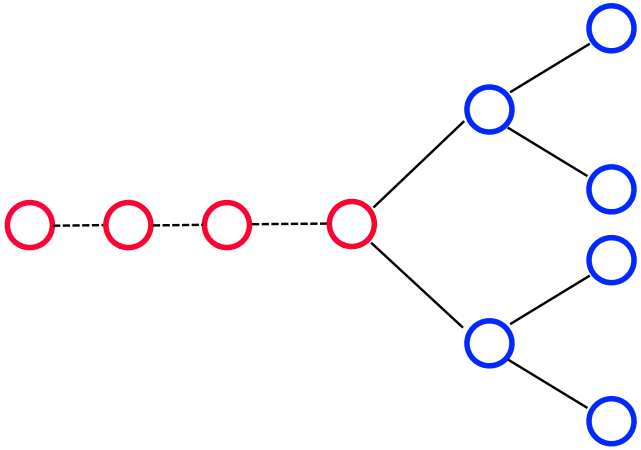
commit if an action in prefix is certainly the best

to gain more planning time for next iteration



Flexible Action Commitment Search

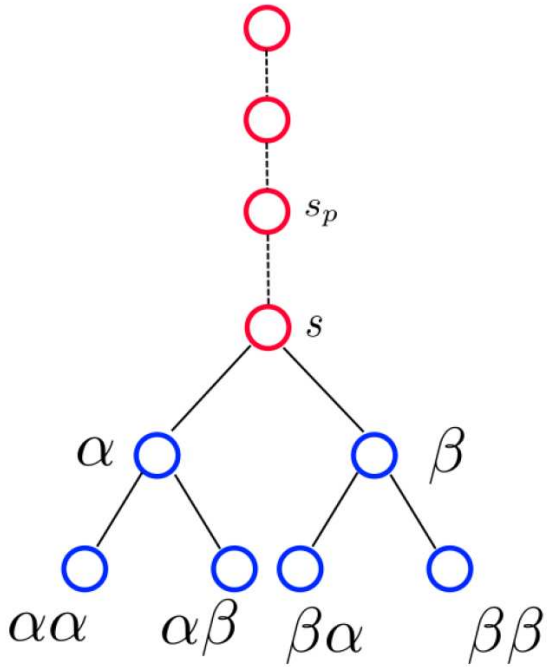
Assumptions

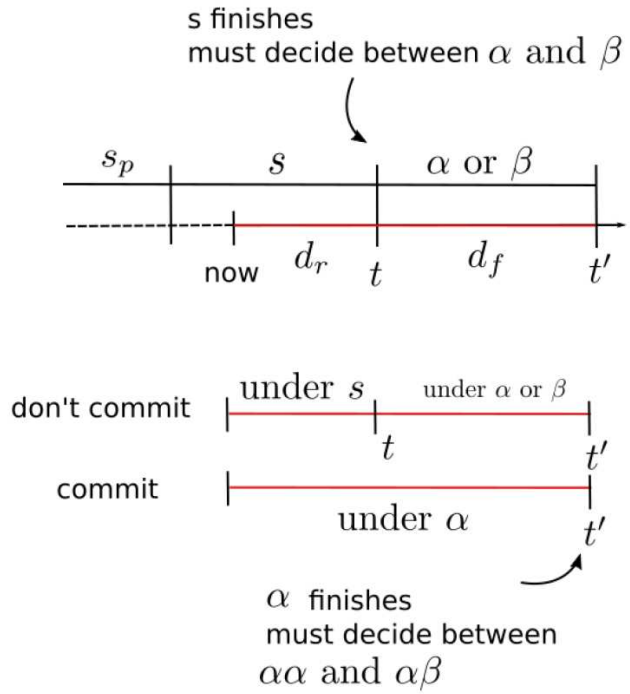


1. system can't be uncontrolled, so force to commit if action queue is empty
2. search tree structure (order of decisions is fixed)
3. no replanning required
4. deterministic system
5. only propose commitment strategy

Our Approach: Flexible Action Commitment Search (FACS)

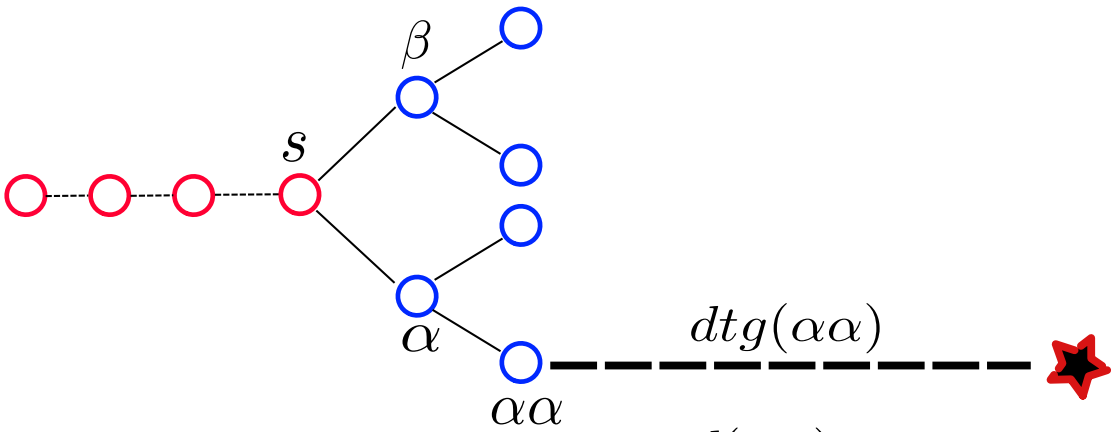
we propose a principled way to make meta-level decision





FACS: The Effect of Search

belief of where $\hat{f}$ will be after search:

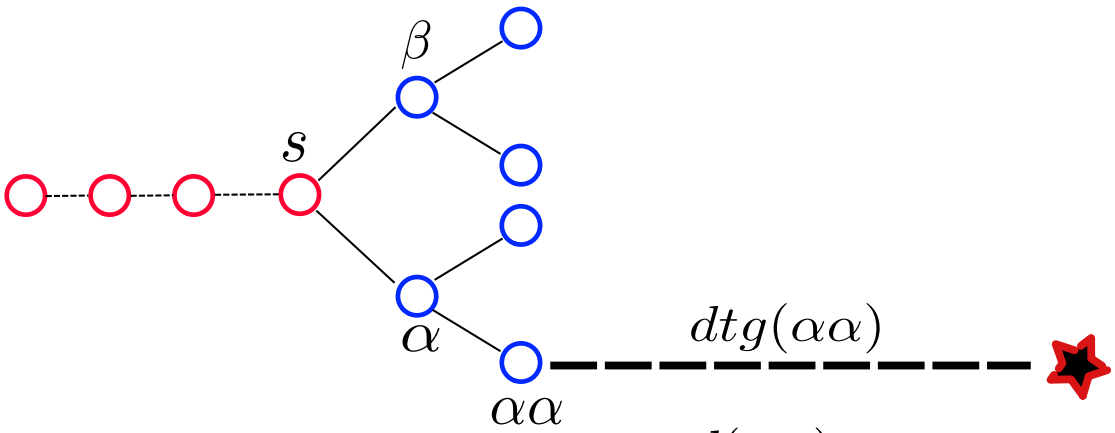


long search duration: $d(\alpha\alpha)$

short search duration: $d(\alpha\alpha)$

FACS: The Effect of Search

belief of where $\hat{f}$ will be after search:



long search duration: $d(\alpha\alpha)$

short search duration: $d(\alpha\alpha)$

$$X^d_{\alpha\alpha} \sim \mathcal{N}(\hat{f}(\alpha\alpha), (\bar{\epsilon}_\alpha \cdot dtg(\alpha\alpha))^2 \cdot \min(1, \frac{\frac{d}{ed}}{dtg(\alpha\alpha)}))$$

FACS: Compute Utility

$$U_{\text{commit}} = \mathbb{E}\left[\min(X^d_{\alpha\alpha}, X^d_{\alpha\beta})\right]$$

where $d = d_r + d_f$

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

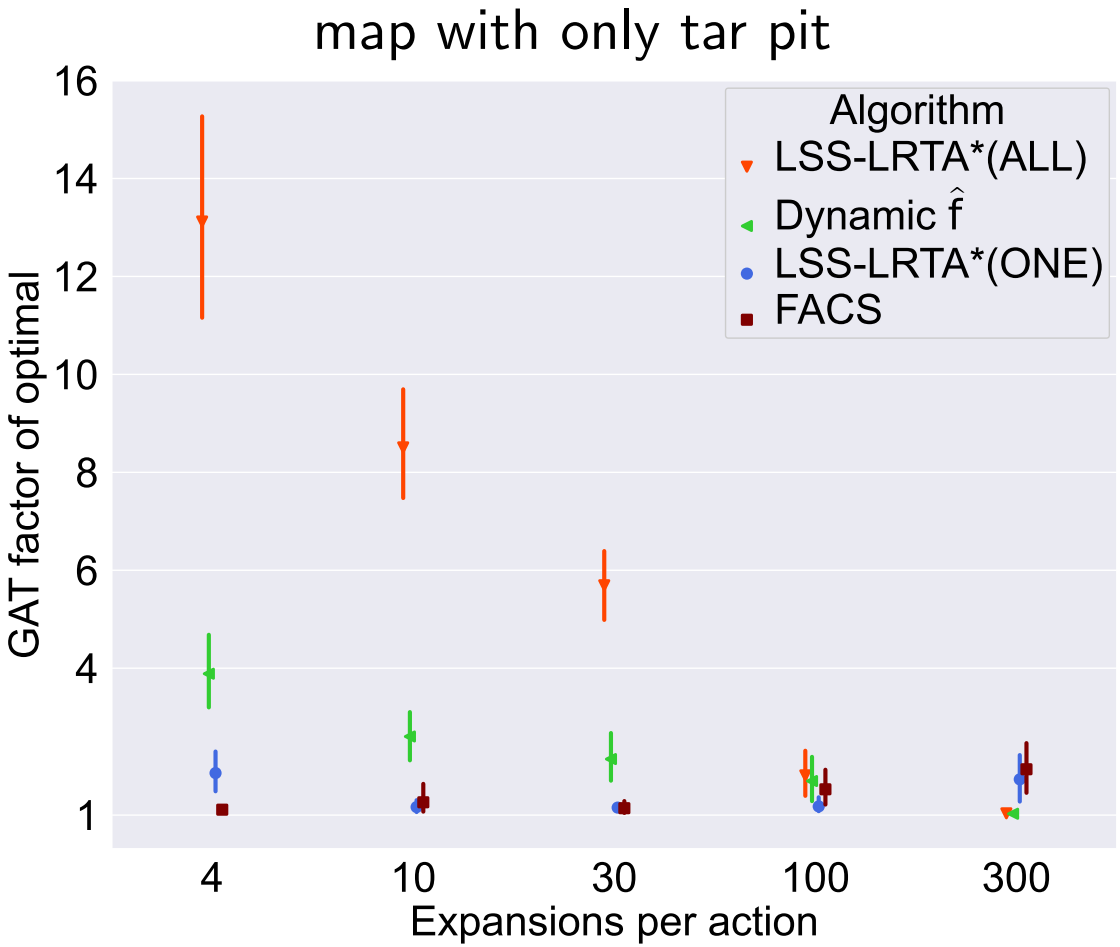
commit when $U^{t'}_{\text{commit}} > U^{t'}_{\text{don't commit}}$

Results

Synthetic Grid Pathfinding

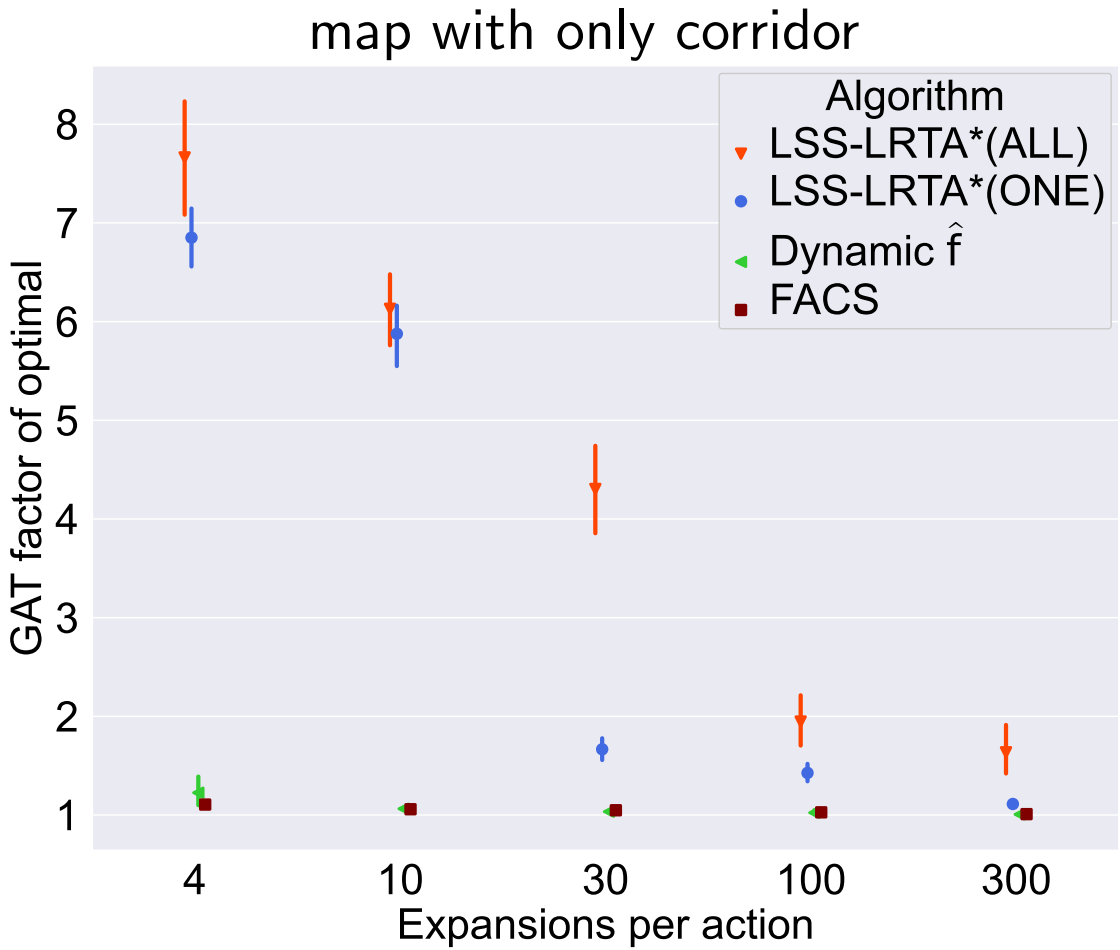
- Left: tar pit area → high cost for reckless committing
- Right: corridor area → need long lookahead to observe the local minima
- Middle: empty area → gain lookahead, no harm to commit

Results



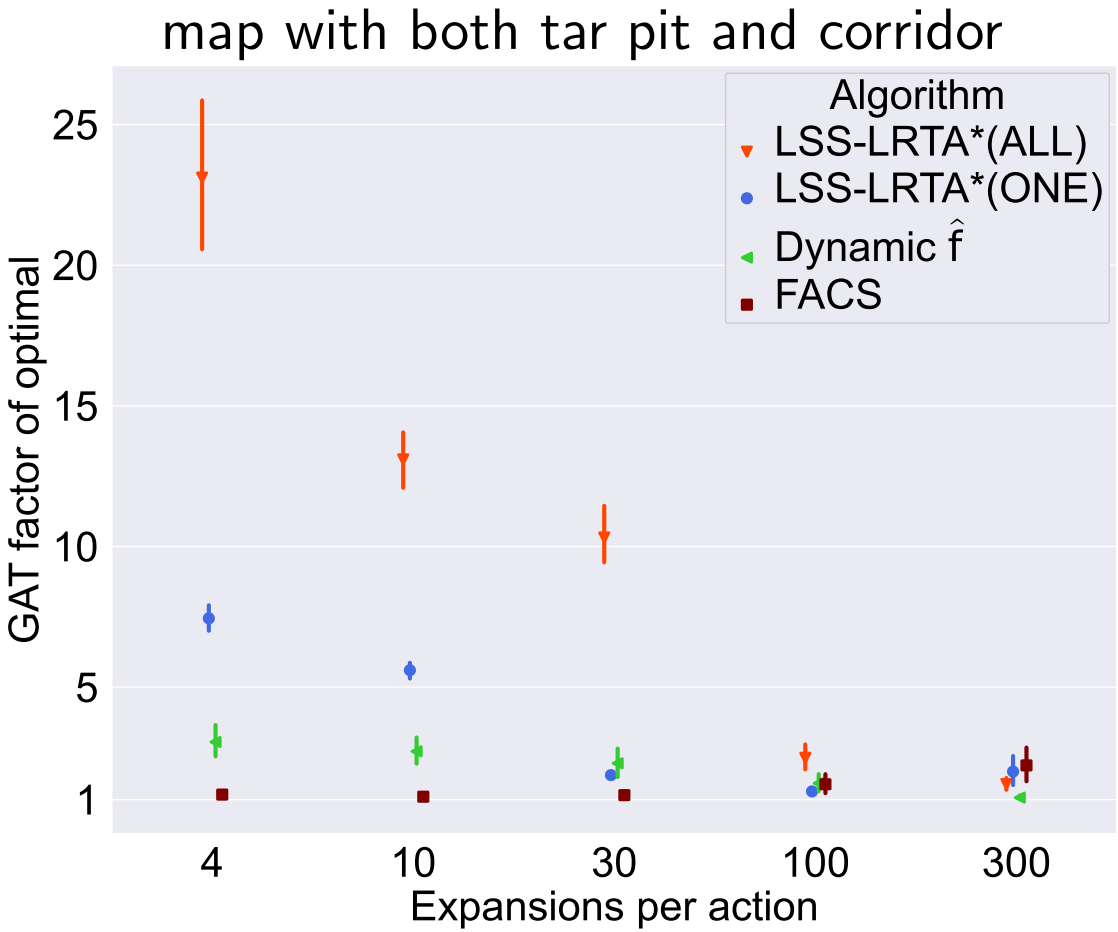
commit-all perform badly

Results



algorithms with small action queue perform badly

Results



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.

More broadly:

- Metareasoning pays off when planning under time pressure!

Questions?