Vectorized Online POMDP Planning

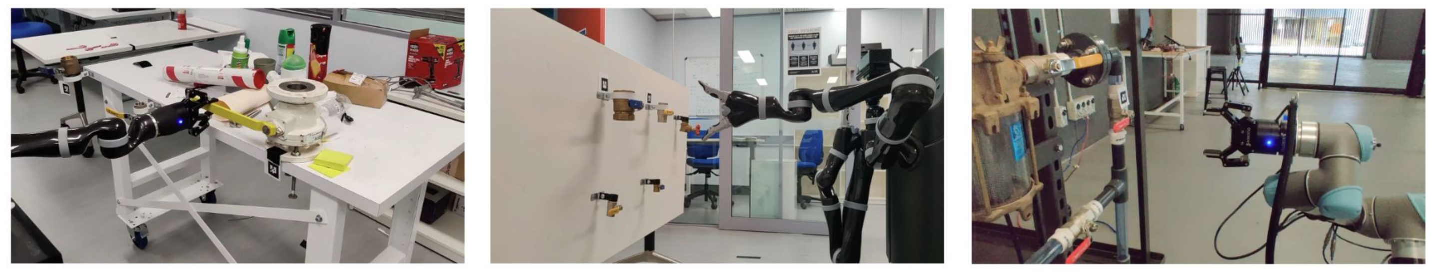
Australian National University | School of Computing

Marcus Hoerger, Muhammad Sudrajat, Hanna Kurniawati

Contact: marcus.hoerger@anu.edu.au

Planning under Uncertainty

Planning under uncertainty is an essential capability of autonomous robots.



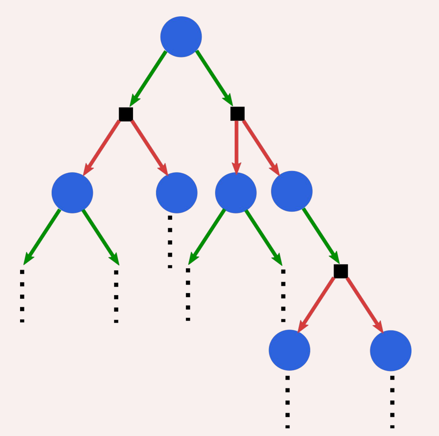
Partially Observable Markov Decision Process (POMDP) provides a general framework for planning under uncertainty.

- **Goal in POMDP planning**: Compute optimal policy that maximises POMDP value function:

$$V(b) = \max_a \left[R(b,a) + \gamma \sum_o p(o \mid b,a) V(b') \right], \quad b = \text{belief}$$

Limitations of existing POMDP solvers

Tree-Search based



- Interleaved numerical optimization & value estimation
- Sequential dependencies between simulations
- Synchronization bottlenecks limit parallelization

Key Ideas

Build on PORPP [1] to **analytically** solve value function, leaving numerical computations for estimation of expectations only:

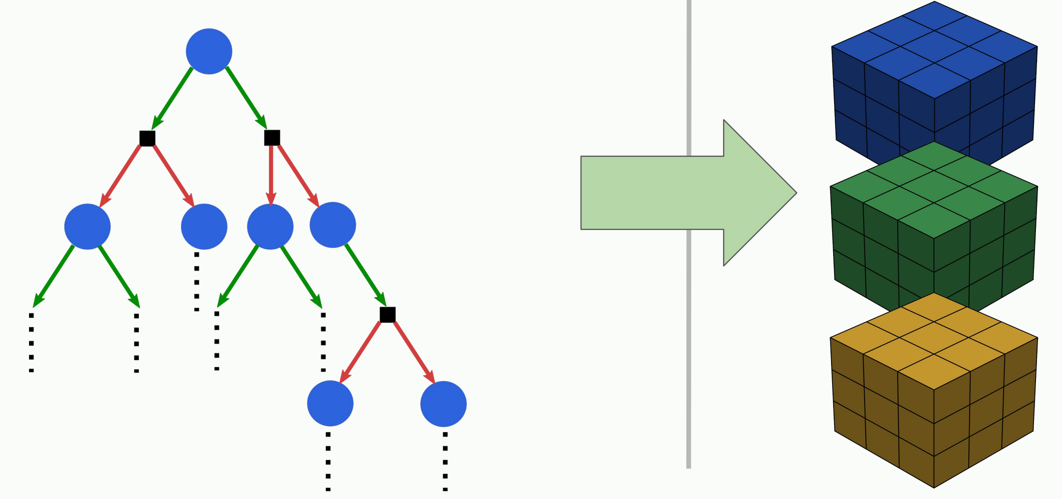
- Analytical value function: $V(b) = \frac{1}{\eta} \sum_{a \in \mathcal{A}} \exp\left[\eta \Psi(b,a)\right]$
- Action preference values Ψ induce policy:

$$\pi(\cdot \mid b) = \mathrm{Softmax}(\Psi(b,\cdot))$$

Represent belief tree as **tensors** and formulate planning as a **fully vectorized** tensor program.

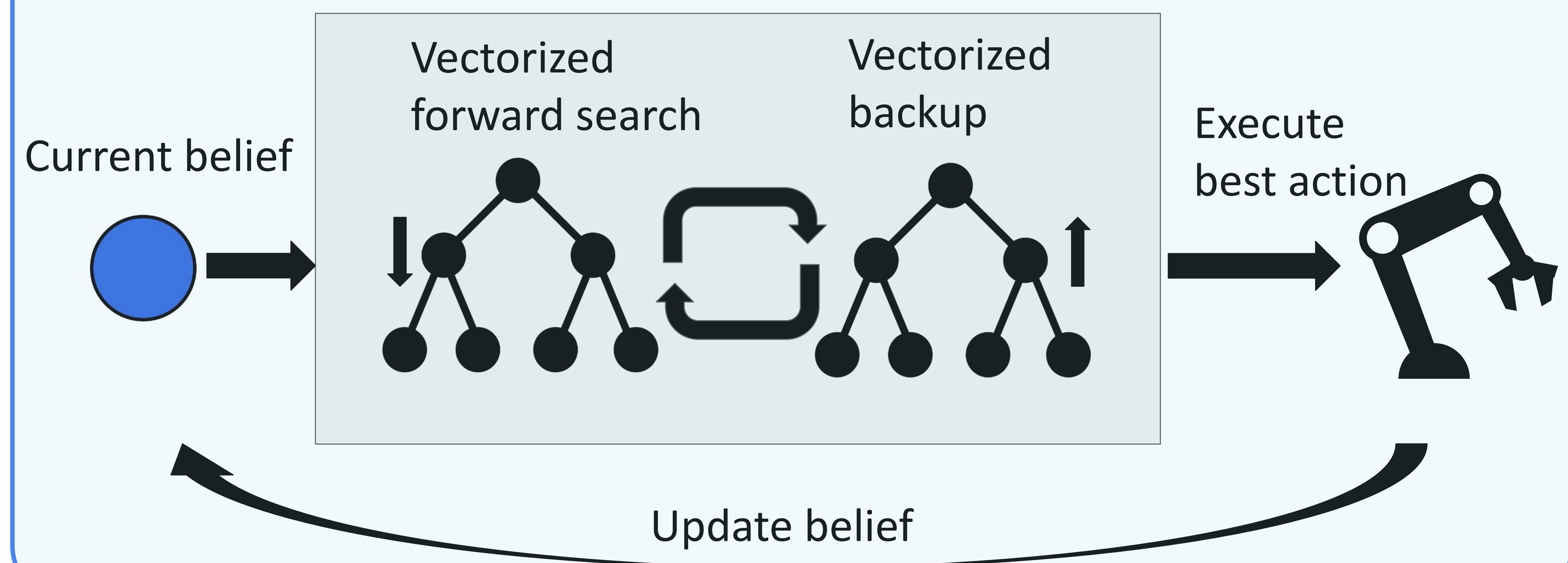
Traditional Solvers

VOPP - Tensorized Belief Tree

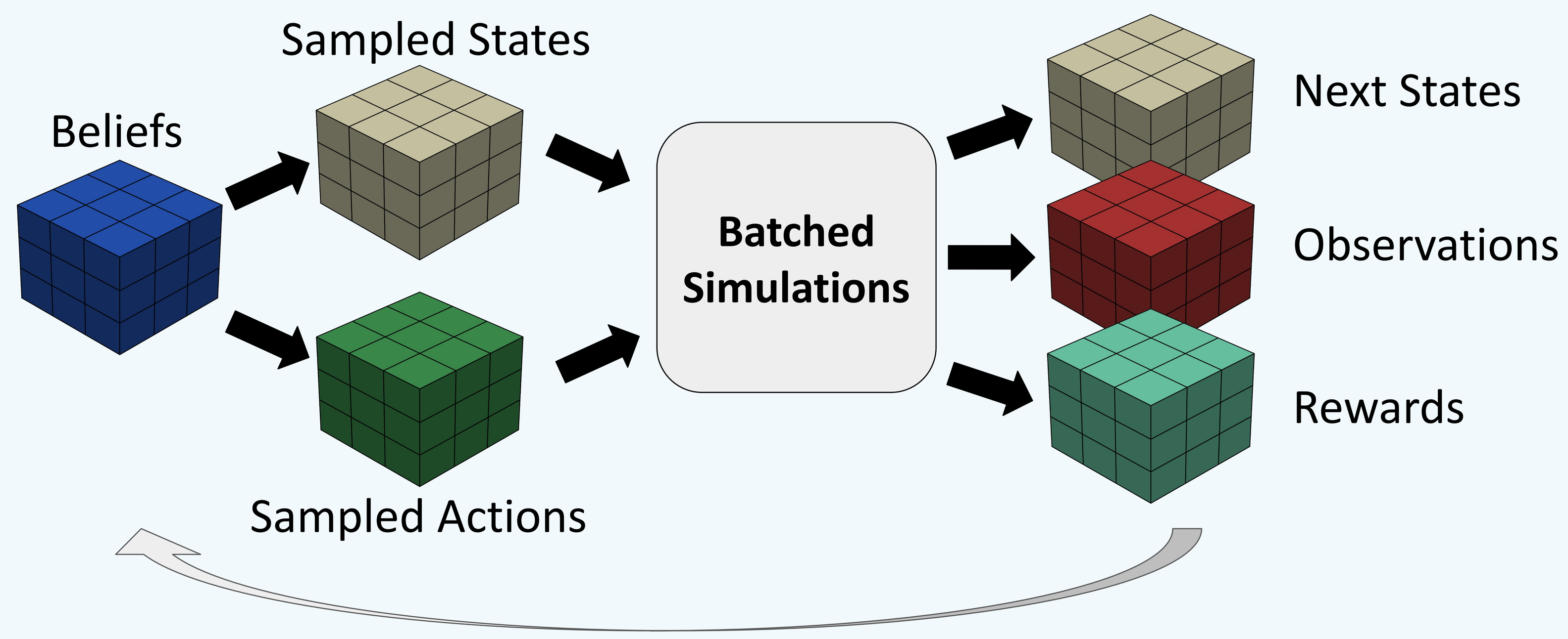


VOPP: Vectorized Online POMDP Planner

VOPP Overview



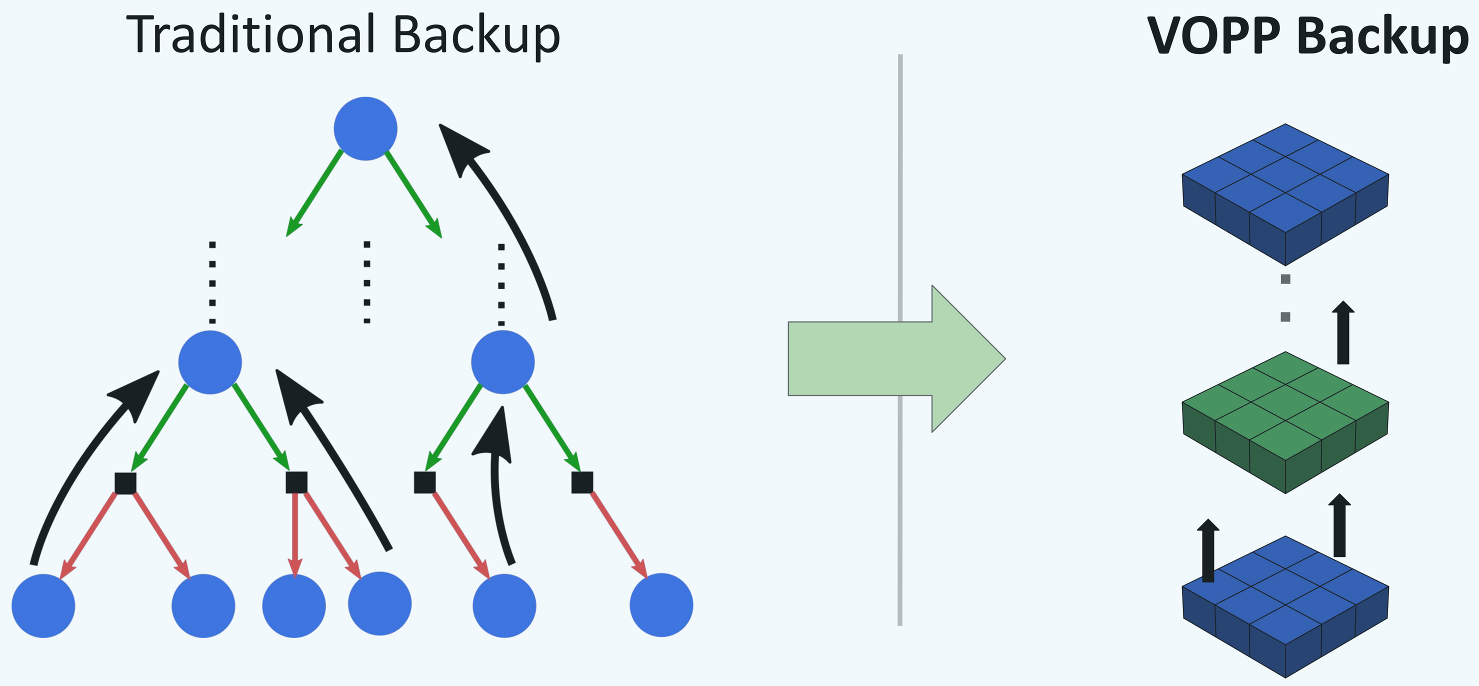
1. Vectorized Forward Search



VOPP performs ***fully vectorized*** forward search to expand the belief tree tensor data structures.

- At each belief b, actions are sampled via $a \sim \mathrm{Softmax}(\Psi(b,\cdot))$ ⟶ embarrassingly parallel

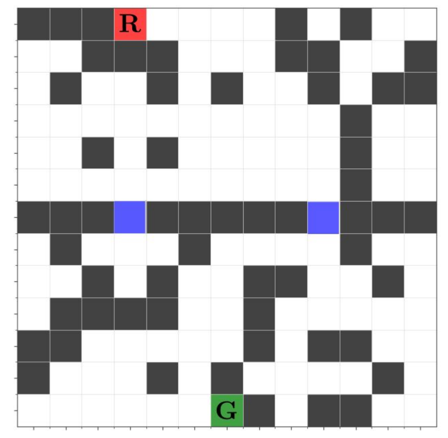
2. Vectorized Backup



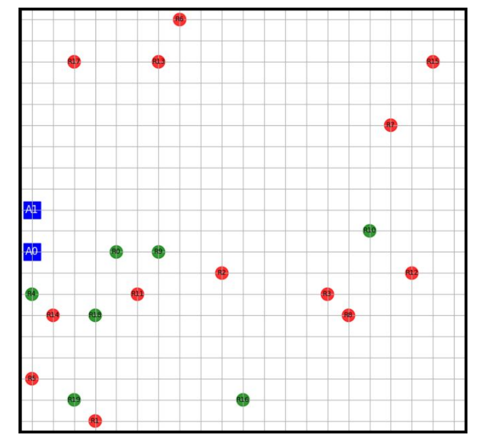
VOPP performs ***fully vectorized*** backup operations to update action preferences Ψ and ***analytically*** compute value functions via $V(b) = \frac{1}{\eta} \sum_{a \in \mathcal{A}} \exp\left[\eta \Psi(b,a)\right]$

Results

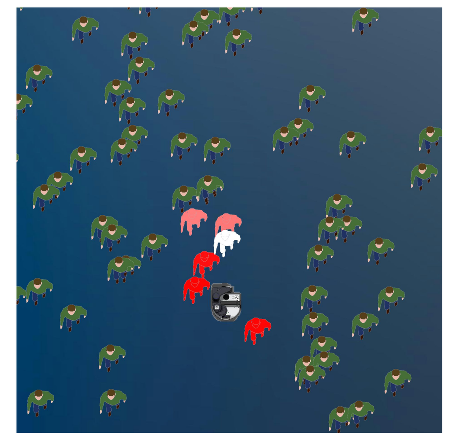
Problem Domains:



Navigation ($|S|$ = 169 x 2^164, $|A|$ = 8, $|O|$ = 8)

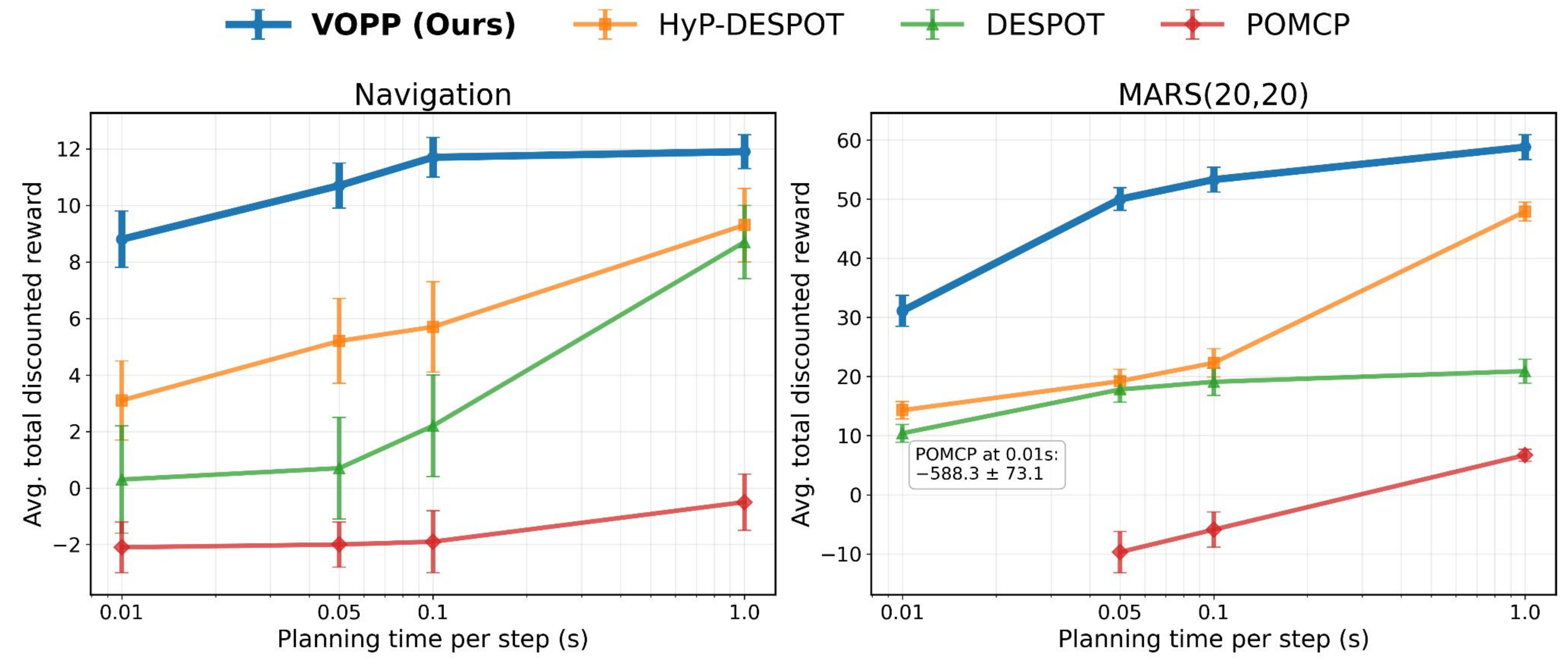


MA-Rocksample (MARS) ($|S|$ = 20^6, $|A|$ = 625, $|O|$ = 9)



CrowdNav (S = R^14 x {0, 1}, $|A|$ = 5, $|O|$ = 64)

Results (additional results in the paper):



MARS(50,50), 3025 actions, 1.0s/step: VOPP achieved 45.1 ± 2.0; other solvers crashed.

- VOPP is at least **20X** more efficient than SOTA parallel method HyP-DESPOT [2].
- VOPP outperforms sequential methods POMCP [3] and DESPOT[4], while using a **1000X** smaller planning budget

References

[1] Kim and Kurniawati. Partially Observable Reference Policy Programming: Approximately Solving POMDPs Sans Numerical Optimisation. IJCAI. 2025.

[2] Cai et al. Hyp-DESPOT: A Hybrid Parallel Algorithm for Online Planning under Uncertainty. IJRR, 2021.

[3] Silver et al. Monte-Carlo Planning in Large POMDPs. NIPS. 2010.

[4] Ye et al. DESPOT: Online POMDP Planning with Regularization. JAIR. 2017.

Paper **Code**