Rule-based Shielding for Partially Observable Monte-Carlo Planning

Giulio Mazzi, Alberto Castellini, Alessandro Farinelli

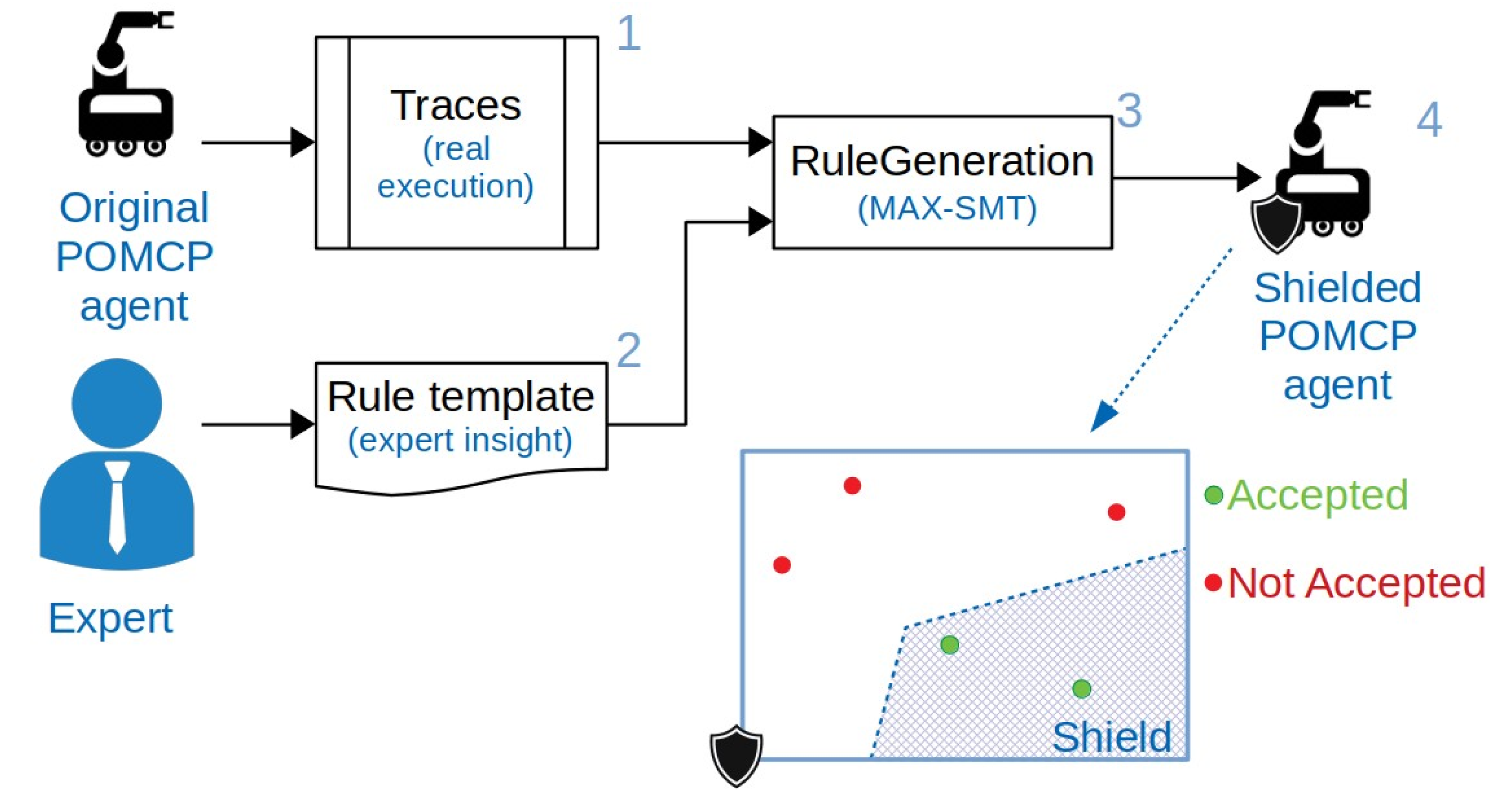
Università degli Studi di Verona, Dipartimento di Informatica

`Giulio.mazzi@univr.it, alberto.castellini@univr.it, alessandro.farinelli@univr.it`

Abstract

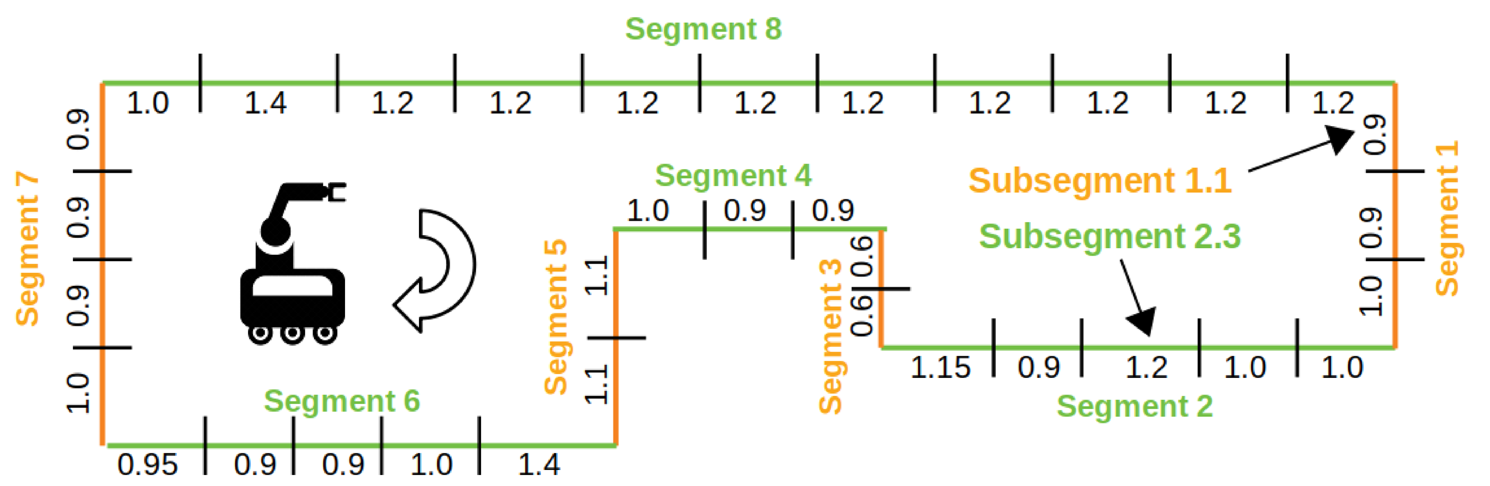
Partially Observable Monte-Carlo Planning (POMCP) is a powerful online algorithm [1]. The online nature of this method supports scalability by avoiding complete policy representation. The lack of an explicit representation however hinders policy interpretability and makes policy verification very complex. In this work, we propose two contributions. The first is a MAX-SMT based method for identifying unexpected actions selected by POMCP with respect to expert prior knowledge of the task [2]. The second is a shielding approach that prevents POMCP from selecting unexpected actions. It identifies anomalous actions selected by POMCP and substitutes those actions with actions that satisfy the logical formulas fulfilling expert knowledge.

Methodology Overview



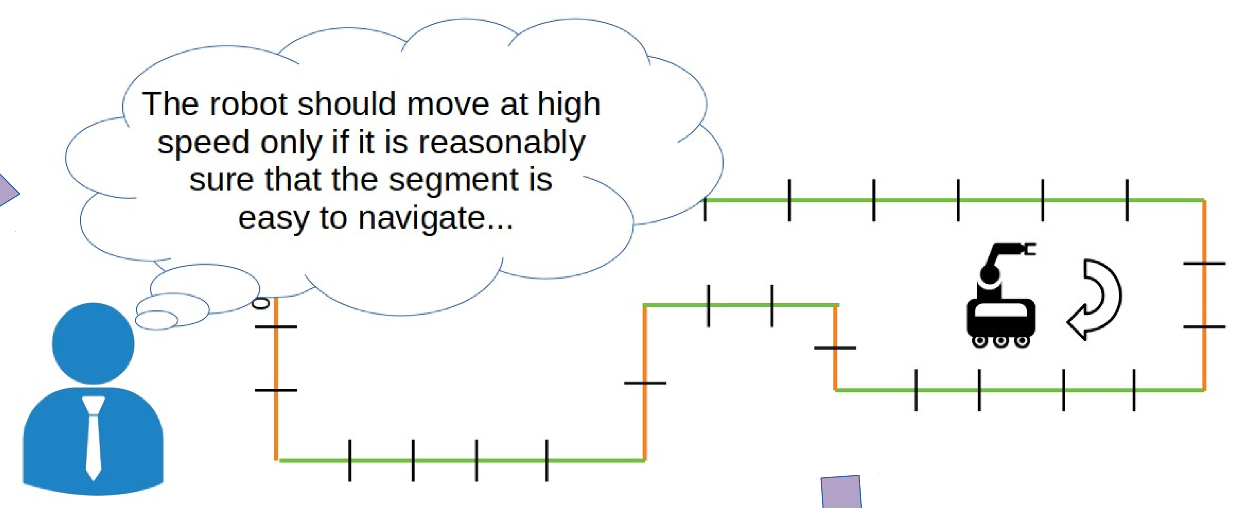
The methodology combines a logic-based high-level insight (1) with an analysis of the execution traces generated by POMCP (2). It synthesizes a rule (3) that is then integrated into a POMCP agent (4) to prevent unwanted behavior online execution.

Shield Synthesis Example



Velocity Regulation: We want a shield that regulates the policy of a mobile robot.

```
Segment:1, Subsegment:1, Difficulty = ( ... ) → A=0
Segment:1, Subsegment:2, Difficulty = ( ... ) → A=2
...
```



We have a high-level requirement for the policy. We formalize it by using a logical formula.

Rule for action move fast

At least one must be satisfied

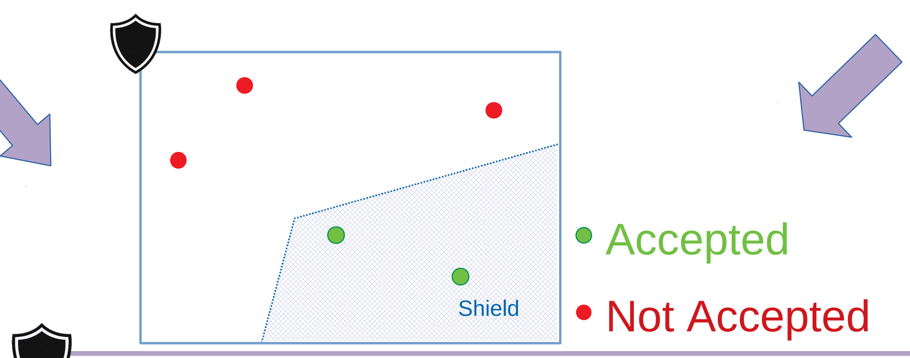
select S_{Fast} when $p(clear) \geq \bar{\mathbf{x}}_1 \vee p(heavy\_cluttered) \leq \bar{\mathbf{x}}_2$

where $\bar{\mathbf{x}}_1 \geq 0.8$

Hard requirements

We want a local rule: we use a function to extract the difficulty of the current segment

We collect the decisions of POMCP into *traces.* We analyze them with a MAX-SMT based algorithm.



We synthesize a shield by combining the execution traces and the logical requirement.

select S_{Fast} when $p(clear) \geq 0.858 \vee p(heavy\_cluttered) \leq 0.004$

Experimental Results

	No Shield		Shield			
c	return	time (s)	return	RI	time (s)	#SA
110	3.702(±0.623)	0.066(±0.027)	3.702(±0.623)	0.00%	0.065(±0.029)	0
80	3.593(±0.632)	0.067(±0.030)	**3.702 (± 0.623)**	3.03%	0.061(±0.027)	4
60	3.088(±0.673)	0.060(±0.025)	**3.702 (± 0.623)**	19.88%	0.061(±0.027)	121
40	−4.173(±1.101)	0.035(±0.017)	**3.702 (± 0.623)**	188.71%	0.052(±0.023)	647

a) Tiger

	No Shield		Shield			
c	return	time (s)	return	RI	time (s)	#SA
103	24.716(±3.497)	10.166(±0.682)	**26.045 (± 3.640)**	5.38%	10.118(±0.238)	7
90	18.030(±3.794)	10.173(±0.234)	**22.680 (± 3.524)**	25.79%	10.166(±0.241)	12
70	4.943(±5.260)	10.278(±0.234)	**8.970 (± 4.556)**	81.46%	10.377(±0.230)	51
50	0.692(±5.051)	10.374(±0.230)	1.638(±4.525)	136.53%	10.435(±0.336)	171

b) Velocity Regulation

A low value of *c* (reward range) generates more errors. The table shows the discounted return, the execution time, the relative increase (RI) in performance, and the shielded actions (#SA). The shielded POMCP outperforms the original algorithm in both *Tiger* and *Velocity Regulation.*

References

[1] Silver, D.; and Veneness, J. Monte-Carlo Planning in Large POMDPs. NeurIPS 2010

[2] Mazzi, G., Castellini, A., Farinelli, A. Identification of Unexpected Decisionsion in Partially Observable Monte Carlo Planning: A Rule-Based Approach. AAMAS 2021

Take Home Message

We presented a safety mechanism for POMCP built from high-level rules. It shields the real-time execution to prevent unexpected decisions.