Advances in SAYNT
Symbiotic Policy Synthesis in POMDPs

Roman Andriushchenko\textsuperscript{1}, \textbf{Alexander Bork}\textsuperscript{2}, Milan Češka\textsuperscript{1}, Sebastian Junges\textsuperscript{3}, Joost-Pieter Katoen\textsuperscript{2}, Fillip Macák\textsuperscript{1}

\textsuperscript{1} Brno University of Technology, CZ
\textsuperscript{2} RWTH Aachen University, DE
\textsuperscript{3} Radboud University Nijmegen, NL
Motivation

Non-deterministic choice
Motivation

Policy resolves non-determinism
Motivation

Probabilistic branching
Motivation

Possible state after action
Motivation — MDP

MDPs are a pivotal model for decision making under uncertainty

Markov Decision Process (MDP)
• Non-deterministic choice
• Probabilistic branching

Rewards
• Used to model steps, costs, …
• Collected when taking a transition

Policy
• Resolves non-determinism
• Maximising/minimising reachability objective:
  Only state-dependent, no memory necessary
Motivation — MDP

MDPs are a pivotal model for decision making under uncertainty.

Markov Decision Process (MDP)
- Non-deterministic choice
- Probabilistic branching

Rewards
- Used to model steps, costs, …
- Collected when taking a transition

Policy
- Resolves non-determinism
- Maximising/minimising reachability objective:
  - Only state-dependent, no memory necessary

Assumes perfect information about state!
Motivation
Motivation

Partial observability
Motivation — POMDP

Partially Observable MDP (POMDP)
• Extension by observation labels

Observation-based policy
• Decisions using observable state information
• Memory is crucial!

POMDPs play an important role for planning in AI
**Policy Synthesis**

- **Goal:** find policy that maximises expected total reward
  - reward collected along all paths until goal state is reached
  - Undiscounted and infinite time horizon

- Optimal policy might not exist
  → Synthesise good policies (Finite State Controllers)
Symbiotic Policy Synthesis

**PAYNT: Inductive Policy Synthesis**
- Synthesise FSC directly
- Use induced MC for value approximation

**STORM: Belief Exploration**
- Construct belief model
- Compute policy using model checking
- Obtain controller from computed policy

**SAYNT: Symbiotic Approach**
Inductive Synthesis for POMDPs — Outer Loop

- **Goal**: learn deterministic FSC
- **Limiting factor**: design space size
- **Access to oracle** can improve design space

![Diagram]

- Learner
- Teacher
- design space (FSCs)
- best FSC

[Andriushchenko et al. 2022]
Inductive Synthesis for POMDPs

Andriushchenko et al. 2022
Inductive Synthesis for POMDPs — Inner Loop

- Teacher gets family of k-FSC
- FSC parameterised in action-choice and memory transitions
- MDP abstraction of family of induced MCs

Compute policy:
(choice of actions + memory structure)

MDP abstraction

Check policy for consistency
(observations)

Consistent

Return FSC

Inconsistent: prune family

[Andriushchenko et al. 2022]
Symbiotic Policy Synthesis

- **PAYNT: Inductive Policy Synthesis**
  - Synthesise a finite state controller
  - Use induced MC for value approximation

- **STORM: Belief Exploration**
  - Construct belief model
  - Compute policy using model checking
  - Obtain controller from computed policy

- **SAYNT: Symbiotic Approach**
POMDP Semantics — Belief MDP

Belief
• Distribution over POMDP states
• Describes likelihood to be in state given observation history

POMDP $\mathcal{M}$

Belief MDP $\text{bel}(\mathcal{M})$

In infinitely many states
Belief Exploration with Cut-Offs

[B., Katoen, Quatmann, 2022]

• Obtain finite MDP for *model checking*
• Explore *part of belief space, approximate values (Cut-Offs)*

• Approximation: based on *some policy* for POMDP

• Weight values by belief distribution, add *goal transition + approx. reward*

\[ R = V(b) \]
Belief Exploration — Example

\[ \sigma_{cut}(s) = \begin{cases} \alpha, & \text{if } O(s) = \bullet \\ \beta, & \text{otherwise} \end{cases} \]
Symbiotic Policy Synthesis

- **PAYNT: Inductive Policy Synthesis**
  - Synthesise finite state controller
  - Use induced MC for value approximation

- **STORM: Belief Exploration**
  - Construct belief model
  - Compute policy using model checking
  - Obtain controller from computed policy

- **SAYNT: Symbiotic Approach**
Symbiotic Approach — Overview

STORM
Compute using belief exploration
Used for cut-offs

PAYNT
Guide exploration of design space
Synthesise

FSC
• Use FSC synthesis for cut-off values
• FSC induces state values
  • Convex combination with belief
  • Maximisation over memory nodes in induced MC
**STORM** → **PAYNT**

- **STORM** provides policy on cut-off MDP = FSC

---

**STORM**

- Memory injection based on choices
- Learner → Teacher
- best FSC

**PAYNT**

- Prioritisation of chosen actions
- Searcher → Eval
- Values & conflicts
- Design space
- Learner
- Teacher
- Best FSC
- Searcher
- Eval
- Values & conflicts

---

Advances in SAYNT - Symbiotic Policy Synthesis in POMDPs
Alexander Bork | LiVe Workshop 2024
Implementation

• Integrated in STORM/PAYNT

• Minimisation/Maximisation
  • Reachability Probabilities
  • Expected Total Rewards

• Part of main releases
### Results — CAV ’23 (Excerpt)

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Over-Arr.</th>
<th>PAYNT</th>
<th>STORM</th>
<th>SAYNT</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Refuel 20 - max</strong></td>
<td>≤ 0.99</td>
<td>0.02</td>
<td>0.15</td>
<td>0.24</td>
</tr>
<tr>
<td>(6834/24k/66)</td>
<td></td>
<td>922s</td>
<td>468s</td>
<td>386s</td>
</tr>
<tr>
<td><strong>Drone 8-2 - max</strong></td>
<td>≤ 0.99</td>
<td>0.9</td>
<td>0.68</td>
<td>0.96</td>
</tr>
<tr>
<td>(13k/32k/3195)</td>
<td></td>
<td>260s</td>
<td>98s</td>
<td>247s</td>
</tr>
<tr>
<td><strong>Netw 3-8-20 - min</strong></td>
<td>≥ 4.31</td>
<td>11.04</td>
<td>10.27</td>
<td>10</td>
</tr>
<tr>
<td>(17k/30k/2205)</td>
<td></td>
<td>638s</td>
<td>238s</td>
<td>742s</td>
</tr>
<tr>
<td><strong>Lanes+ - min</strong></td>
<td>≥ 4805</td>
<td>8223</td>
<td>18870</td>
<td>4805</td>
</tr>
<tr>
<td>(2741/5289/11)</td>
<td></td>
<td>118s</td>
<td>376s</td>
<td>173s</td>
</tr>
<tr>
<td><strong>4x5x2 95 - max</strong></td>
<td>≤ 3.26</td>
<td>0.94</td>
<td>2.08</td>
<td>2.08</td>
</tr>
<tr>
<td>(79/310/7)</td>
<td></td>
<td>305s</td>
<td>3s</td>
<td>71s</td>
</tr>
</tbody>
</table>

*max: larger is better*

*min: smaller is better*

Better result, sometimes faster

No improvement

Lots of additional results in the paper
Advances — Focused FSC Synthesis

• **Motivation:**
  - **PAYNT** - optimise FSC for initial state
  - good in initial state ≠ good for all beliefs

• Seed synthesis in **cut-off beliefs**
• Use **over-approximation** as guide
• Prioritise **large gaps**
Focused FSC Synthesis — Prelim. Results

**BUT:** STORM’s over-approximations are costly for small benefit

→ better over-approximations?

![Graphs showing time vs. value for Milos-97 and Refuel-20]
Advances — Discounting

- **Discounted reward:** standard in AI applications
- Solvers available (**SARSOP** [Kurniawati, Hsu, Lee 2008], …)

- Added support in **STORM**
  - Modify MDP model checking engine
  - enables discounting in **PAYNT** and **STORM** POMDP

*No results to report yet, stay tuned!*
Conclusion

Policy Synthesis in POMDPs
- Difficult problem, practically relevant
- Approximation necessary

Our Approach SAYNT
- Inductive synthesis + belief exploration
- Experiments show potential of symbiosis

Current Developments
- Multiple FSCs
- Integration of over-approximations
- Discounting

Scan for CAV ’23 Paper