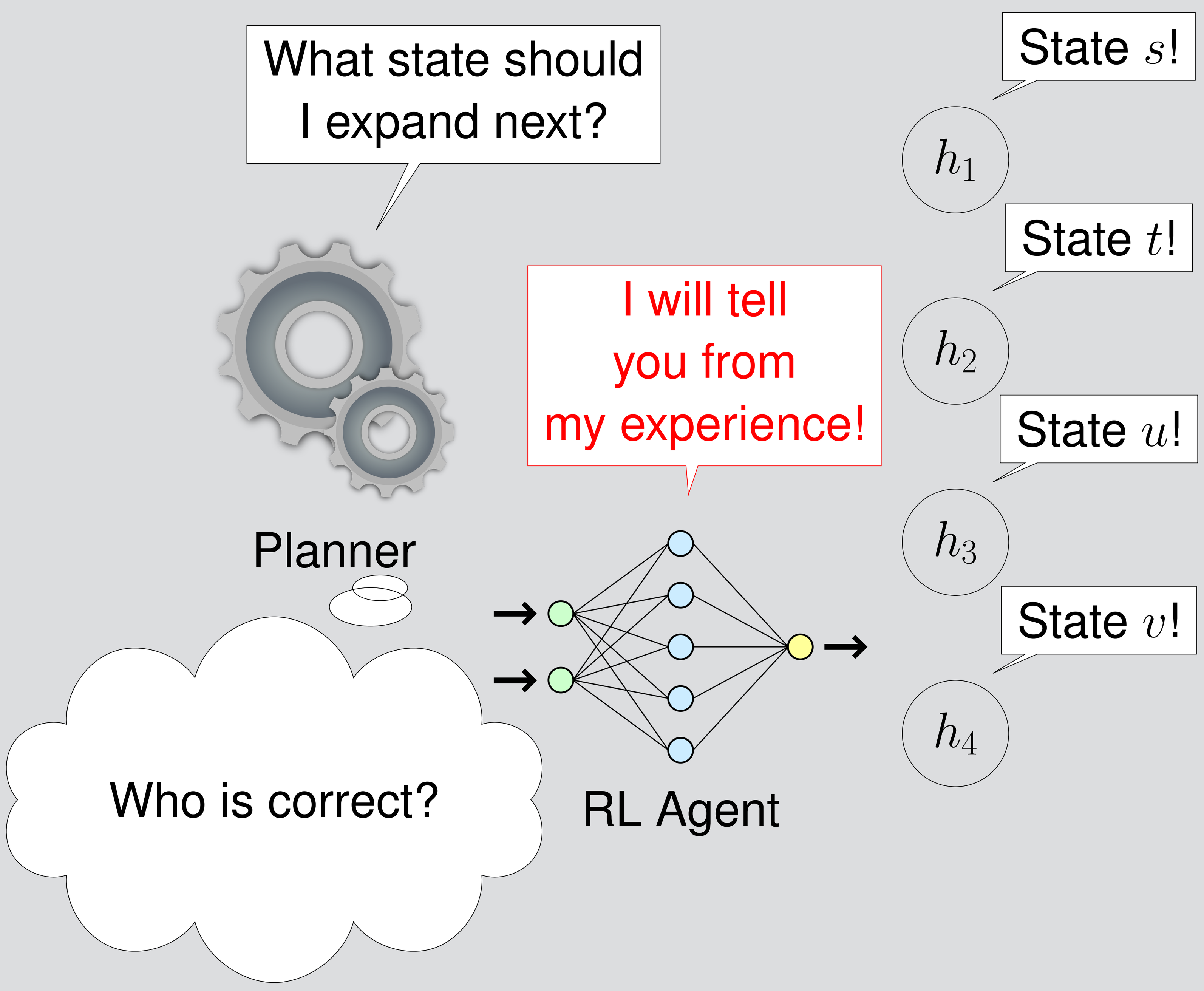
Learning Heuristic Selection with Dynamic Algorithm Configuration

David Speck[1], André Biedenkapp[1], Frank Hutter[1,2],
Robert Mattmüller[1] and Marius Lindauer[3]
⟨speckd, biedenka, fh, mattmuel⟩@informatik.uni-freiburg.de, lindauer@tnt.uni-hannover.de
[1]University of Freiburg, [2]Bosch Center for Artificial Intelligence, [3]Leibniz University Hannover

UNI FREIBURG

Motivation

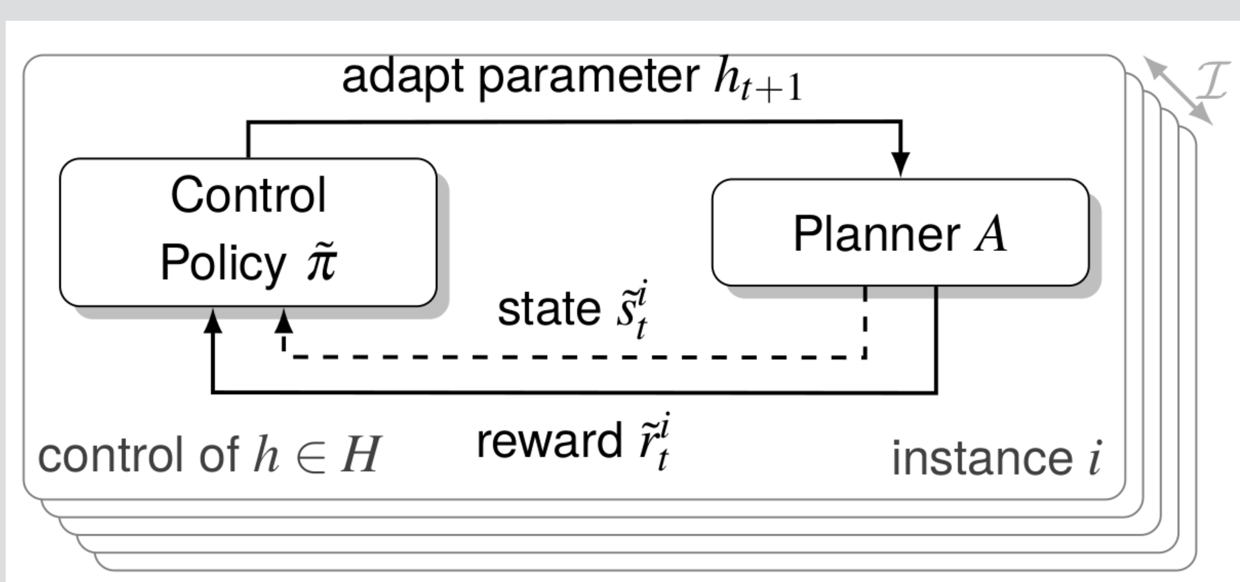


Satisficing planning

- Search for a good plan
- Inadmissible heuristics are difficult to combine
- Greedy search with multiple heuristics
  - States evaluated with each heuristic
  - One separate open list for each heuristic

Automated Algorithm Configuration

- Algorithm Selection $\tilde{\pi} : \mathcal{I} \to H$
  - Considers instance
  - E.g. portfolio planner
- Adaptive Algorithm Configuration $\tilde{\pi} : \mathbb{N}_0 \to H$
  - Considers time step
  - E.g. alternation between heuristics
- Dyn. Algorithm Configuration $\tilde{\pi} : \mathcal{I} \times \mathbb{N}_0 \times \tilde{\mathcal{S}} \to H$
  - Considers instance, time step and planner state
  - Problem can be considered as MDP
  - Our approach based on Reinforcement Learning



Dynamic Algorithm Configuration – Theoretical Properties

- An optimal DAC policy is at least as good as an optimal AS policy and an optimal AAC policy. □
- There is a family of planning tasks so that a DAC policy expands exponentially fewer states until a plan is found. □

Features and Rewards

- Features for each heuristic $h \in H$ (open list)
  - $\max_h$, $\min_h$, μ_h, σ^2_h, $\#_h$ and $t \in \mathbb{N}_0$
- Difference of each feature between $t-1$ and t
- Reward: -1 for each expansion step until solution is found

Experiments

- $H = \{h_{\text{ff}}, h_{\text{cg}}, h_{\text{cea}}, h_{\text{add}}\}$
- 6 domains with 100 instances per train/test Set
- ϵ-greedy deep Q-learning (double DQN)
  - 2-layer network with 75 hidden units
  - 5 different DAC polices per domain

Algorithm	CONTROL POLICY			SINGLE HEURISTIC				BEST AS
Domain (# Inst.)	RL	RND	ALT	h_{ff}	h_{cg}	h_{cea}	h_{add}	SGL. h
BARMAN (100)	**84.4**	83.8	83.3	66.0	17.0	18.0	18.0	67.0
BLOCKS (100)	**92.9**	83.6	83.7	75.0	60.0	92.0	92.0	93.0
CHILDS (100)	**88.0**	86.2	86.7	75.0	86.0	86.0	86.0	86.0
ROVERS (100)	95.2	**96.0**	**96.0**	84.0	72.0	68.0	68.0	91.0
SOKOBAN (100)	87.7	87.1	87.0	88.0	**90.0**	60.0	89.0	92.0
VISITALL (100)	56.9	51.0	51.5	37.0	**60.0**	**60.0**	**60.0**	60.0
SUM (600)	**505.1**	487.7	488.2	425.0	385.0	384.0	413.0	489.0

- Our approach based on RL performs overall best
- Best Algorithm Selection (Oracle) is worse than control policies

Conclusion and Future Work

DAC can improve heuristic selection.

- Considers instance, time step and planner state
- Can improve search performance exponentially
- It is possible to learn good policies
- Future: Investigate domain-specific state features