Understanding Sample Generation Strategies for Learning Heuristic Functions in Classical Planning

Author: Rafael Vales Bettker
Advisor: Prof. Dr. André Grahl Pereira
Co-advisor: Prof. Dr. Marcus Ritt

Federal University of Rio Grande do Sul (UFRGS)
Institute of Informatics
Department of Theoretical Informatics

Table of contents

Introduction

Introduction

- Classical planning provides a method for representing and solving various problems.
- Planning tasks are typically defined by the initial state and the desired outcome (goal state).
- Heuristic functions guide search algorithms to find solutions.
- Learning heuristic functions with neural networks (NN) based on samples that are states with their cost-to-goal estimates.
  - The NN is our heuristic function and solves distinct initial states of a state space.

Problem and Contributions

Learn effective NN-based heuristic functions.

- How to generate good samples?
- What is relevant to the quality of the samples?

Problem and Contributions

Learn effective NN-based heuristic functions.

- How to generate good samples?
- What is relevant to the quality of the samples?

Our contributions include

- A systematic study of sampling strategies.
- New sampling and *h*-value refinement algorithms.

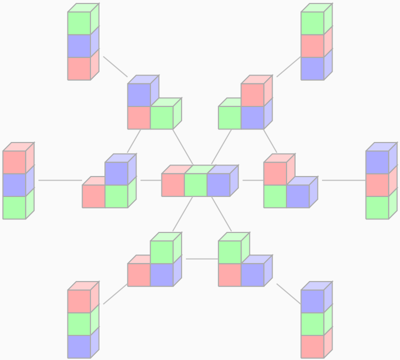
Background

Classical Planning

A planning task can be defined as a tuple $\Pi = \langle \mathcal{S}, \mathcal{O}, s_0, s^*, \text{cost} \rangle$.

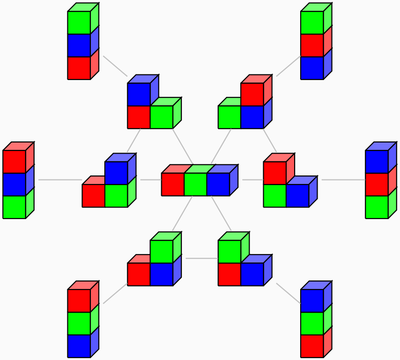
Classical Planning

A planning task can be defined as a tuple $\Pi = \langle \mathcal{S}, \mathcal{O}, s_0, s^*, \text{cost} \rangle$.



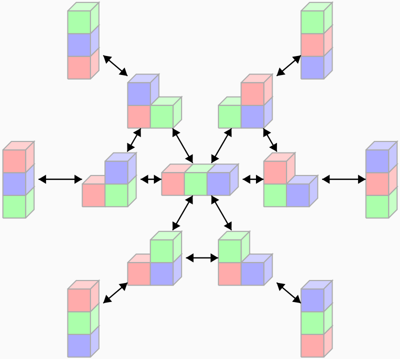
Classical Planning

A planning task can be defined as a tuple $\Pi = \langle \mathcal{S}, \mathcal{O}, s_0, s^*, \text{cost} \rangle$.



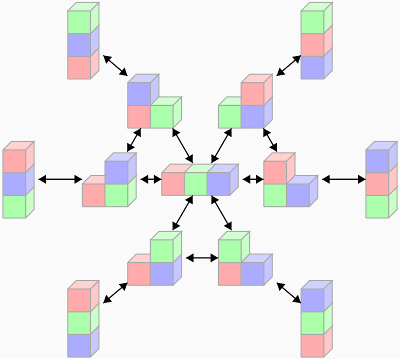
Classical Planning

A planning task can be defined as a tuple $\Pi = \langle \mathcal{S}, \mathcal{O}, s_0, s^*, \mathrm{cost} \rangle$.



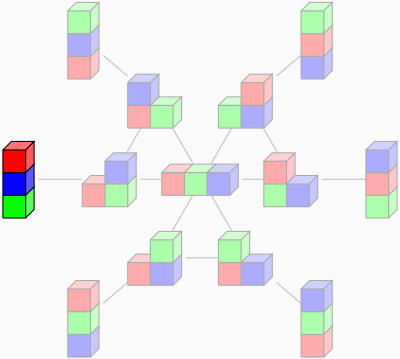
Classical Planning

A planning task can be defined as a tuple $\Pi = \langle \mathcal{S}, \mathcal{O}, s_0, s^*, \text{cost} \rangle$.



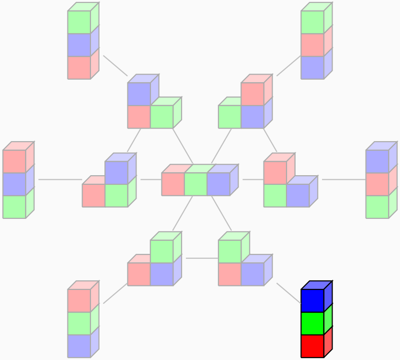
Classical Planning

A planning task can be defined as a tuple $\Pi = \langle \mathcal{S}, \mathcal{O}, s_0, s^*, \text{cost} \rangle$.



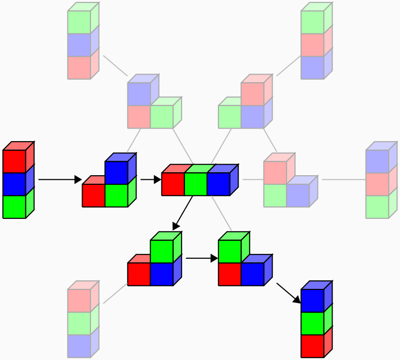
Classical Planning

A planning task can be defined as a tuple $\Pi = \langle \mathcal{S}, \mathcal{O}, s_0, s^*, \text{cost} \rangle$.



Classical Planning

A planning task can be defined as a tuple $\Pi = \langle \mathcal{S}, \mathcal{O}, s_0, s^*, \mathrm{cost} \rangle$.



Related Work

Structured NN-based

- Structured neural networks.
- Encode part of the model into the network architecture.
- Tipically multi-domain or domain-independent.
- E.g. Shen *et al.* (2020)

Non-structured NN-based

- Feedforward neural networks.
- Highly independent of the task description.
- Typically state space-specific.
- E.g. O'Toole *et al.* (2022)

Sample Generation

Sample Generation

- Sample generation is an algorithmic problem.
  - Black-box model: explores the state space through the generation of successors/predecessors.

- Main approaches:
  - Progression: forward search from the initial state.
  - Regression: backward search from the goal state.
  - Random sampling: random generation of states from the state space.

Sampling by Regression

- Expand the backward state space through reverse operators.
- Usually the run is restarted after reaching the regression limit. Each run is called a rollout.
- The cost-to-goal estimates (h-value) of a state is the sum of the operator costs applied since the goal state.
- Main sampling algorithms:
  - Breadth-first Search (BFS)
  - Depth-first Search (DFS)
  - Random Walk (RW)

FSM

Our proposal: FSM

- Combines good coverage at short distances (BFS) with medium to long distances (RW).

Two-stage sampling

1. Generates a portion of the samples in a BFS rollout.
2. Run multiple RW rollouts starting from BFS' leaves.

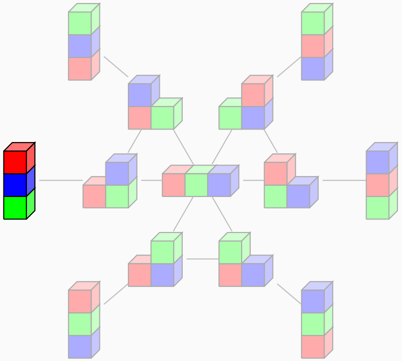
FSM

Our proposal: FSM

- Combines good coverage at short distances (BFS) with medium to long distances (RW).

Two-stage sampling

1. Generates a portion of the samples in a BFS rollout.
2. Run multiple RW rollouts starting from BFS' leaves.



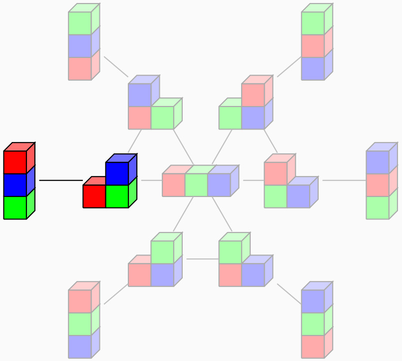
FSM

Our proposal: FSM

- Combines good coverage at short distances (BFS) with medium to long distances (RW).

Two-stage sampling

1. Generates a portion of the samples in a BFS rollout.
2. Run multiple RW rollouts starting from BFS' leaves.



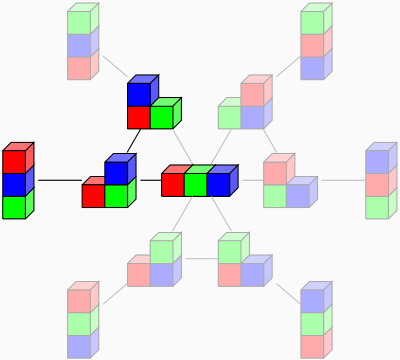
FSM

Our proposal: FSM

- Combines good coverage at short distances (BFS) with medium to long distances (RW).

Two-stage sampling

1. Generates a portion of the samples in a BFS rollout.
2. Run multiple RW rollouts starting from BFS' leaves.



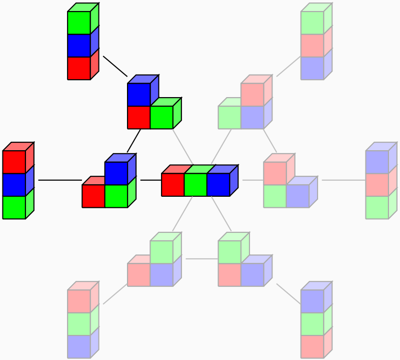
FSM

Our proposal: FSM

- Combines good coverage at short distances (BFS) with medium to long distances (RW).

Two-stage sampling

1. Generates a portion of the samples in a BFS rollout.
2. Run multiple RW rollouts starting from BFS' leaves.



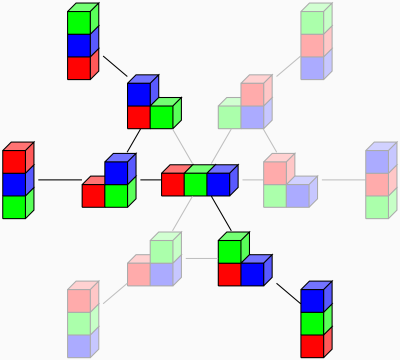
FSM

Our proposal: FSM

- Combines good coverage at short distances (BFS) with medium to long distances (RW).

Two-stage sampling

1. Generates a portion of the samples in a BFS rollout.
2. Run multiple RW rollouts starting from BFS' leaves.

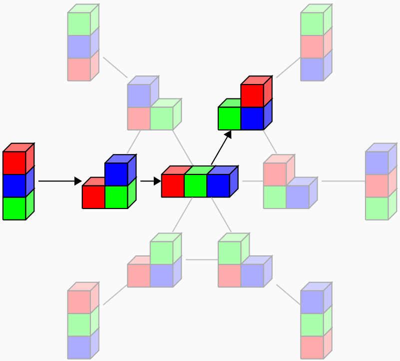


Regression Limit

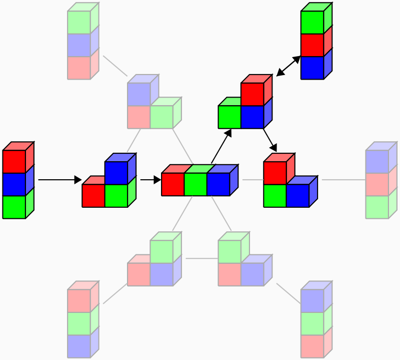
When to stop a rollout?

- Stop after reaching some maximum limit L of samples.
  - Yu *et al.* (2020); O'Toole *et al.* (2022) use fixed maximum limit, 200 and 500 (resp.).
- A fixed maximum limit L is not a good choice for regression sampling.
  - Planning tasks vary in state space size and maximum distance d^* from any state to a goal state.

Regression Limit



If L underestimates d^*



If L overestimates d^* by much

Adaptative Regression Limit

The ideal approach is a regression limit based on d^*.

Our proposals: estimates the d^*-value from task information.

Number of facts

$$F = |\mathcal{F}(s_0)|$$

Number of facts per mean number of effects in the operators

$$\bar{F} = \left\lceil \frac{|\mathcal{F}(s_0)|}{\sum_{o \in \mathcal{O}} \frac{|\mathrm{eff}(o)|}{|\mathcal{O}|}} \right\rceil$$

Randomly Generated Samples

- Regression may have difficulty reaching certain regions.
- Randomly generated samples (random samples) provide uniform coverage across the entire state space.

- Adding random samples to the set of samples improves the performance of the learned heuristic (O'Toole *et al.*, 2022).
  - Generate a random state s with cost-to-goal estimate $h(s) = max(H) + 1$, where H is the set of all cost-to-goal estimates in the original sample set.

Cost-to-goal Estimates

Cost-to-goal Estimates

Based solely on the sampled rollout.

How to improve?

- Use knowledge of all rollouts to update the sampled cost-to-goal estimate.

Sample Improvement

- A same state can be sampled by more than one rollout.
- Repeated states can have different cost-to-goal estimates.
  - Misinformation for NN learning.

Our proposal: for each sampled state s, update its cost-to-goal estimate to $h(s) = \min\{h_i \mid s = s_i, i \in [N]\}$.

- Never underestimate the perfect estimate h^*.
- This approach is called sample improvement (SAI).

Successor Improvement

Multiple rollouts can generate neighboring (a distance operator) states in state space.

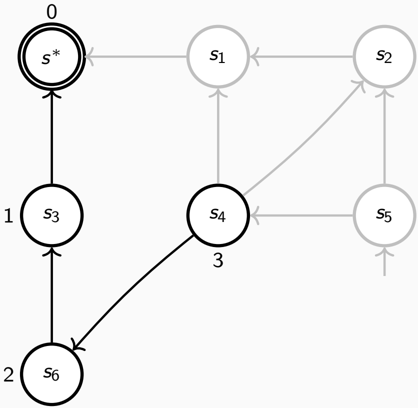
Our proposal: Connect sampled states that are one operator away to create paths that generate knowledge to improve cost-to-goal estimates.

- This approach is called successor improvement (SUI).

Successor Improvement

Rollout #1

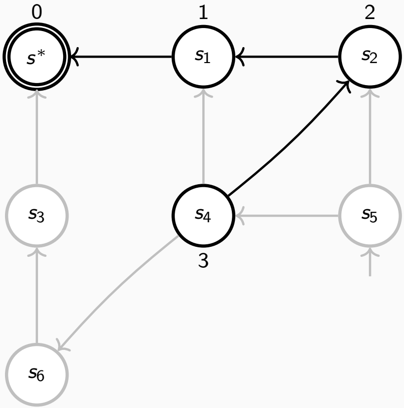
- Sample #1: $\langle s^*, 0\rangle$
- Sample #2: $\langle s_3, 1\rangle$
- Sample #3: $\langle s_6, 2\rangle$
- Sample #4: $\langle s_4, 3\rangle$



Successor Improvement

Rollout #2

- Sample #5: $\langle s^*, 0\rangle$
- Sample #6: $\langle s_1, 1\rangle$
- Sample #7: $\langle s_2, 2\rangle$
- Sample #8: $\langle s_4, 3\rangle$



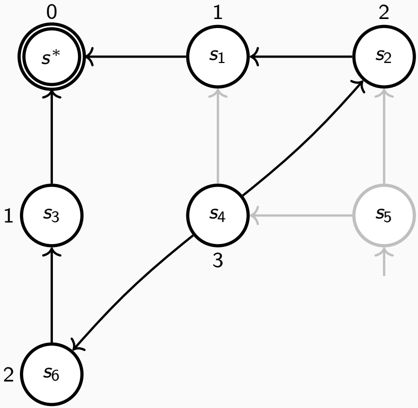
Successor Improvement

Rollout #1

- Sample #1: $\langle s^*, 0\rangle$
- Sample #2: $\langle s_3, 1\rangle$
- Sample #3: $\langle s_6, 2\rangle$
- Sample #4: $\langle s_4, 3\rangle$

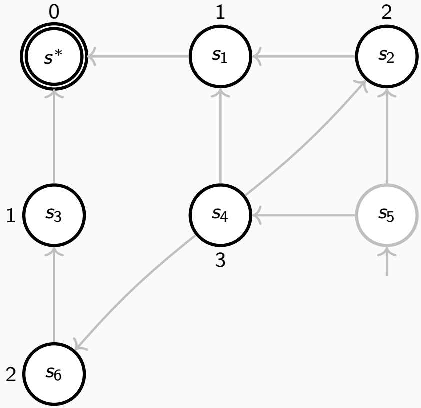
Rollout #2

- Sample #5: $\langle s^*, 0\rangle$
- Sample #6: $\langle s_1, 1\rangle$
- Sample #7: $\langle s_2, 2\rangle$
- Sample #8: $\langle s_4, 3\rangle$



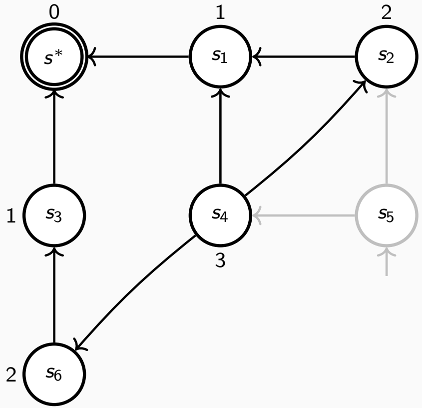
Successor Improvement

1. Consider a directed graph $G = (V, A)$ over all sampled states, i.e., $V = \{s_i \mid i \in [N]\}$.
2. For every pair of states $s, t \in V$ such that for some operator $o \in \mathcal{O}$ applicable to s we have $\text{succ}(s, o) \subseteq t$, we add an arc (s, t) of length $\text{cost}(o)$ to A.
3. Propagates the cost-to-goal estimate of each state to its predecessors.



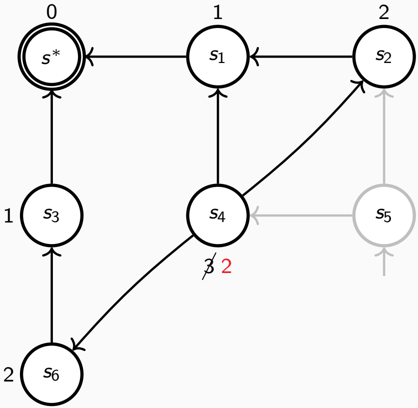
Successor Improvement

1. Consider a directed graph $G = (V, A)$ over all sampled states, i.e., $V = \{s_i \mid i \in [N]\}$.
2. For every pair of states $s, t \in V$ such that for some operator $o \in \mathcal{O}$ applicable to s we have $\text{succ}(s, o) \subseteq t$, we add an arc (s, t) of length $\text{cost}(o)$ to A.
3. Propagates the cost-to-goal estimate of each state to its predecessors.



Successor Improvement

1. Consider a directed graph $G = (V, A)$ over all sampled states, i.e., $V = \{s_i \mid i \in [N]\}$.
2. For every pair of states $s, t \in V$ such that for some operator $o \in \mathcal{O}$ applicable to s we have $\text{succ}(s, o) \subseteq t$, we add an arc (s, t) of length $\text{cost}(o)$ to A.
3. Propagates the cost-to-goal estimate of each state to its predecessors.



Experiments

Experiments

Systematic analysis of the influence of each feature.

- What is the contribution of the sampling algorithm?
- What is the impact of the regression limit?
- How much do random samples improve the sample set?
- How much do our *h*-value improvement techniques refine the estimators?
- What is the quality of the *h*-values produced by the learned heuristics?
- Does the quality of the *h*-values reflect in the search?

The experiments were divided into two parts: small state spaces and large state spaces.

Common Settings

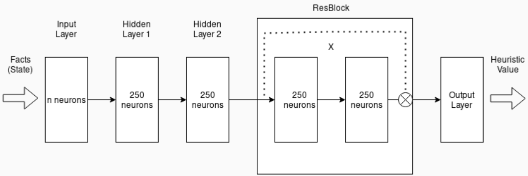


Figure 1: Residual network.

- Same neural network architecture as Ferber *et al.* (2022) and O'Toole *et al.* (2022).
- Baseline $\hat{h}_0$: Random walk sampling, regression limit $L = 200$, and without h-value improvement techniques.
- FSM samples 10 % of the samples with BFS and the rest with RW.
- Dataset from Ferber *et al.* (2022) moderate tasks.
- Each domain: N tasks $\times$ 25 models $\times$ 50 initial states.

Small State Spaces

- Tasks where the forward state space can be enumerated.
    - Allows the generation of all states in the state space and their perfect cost-to-goal estimates (h^*).
    - Better control and understanding of the behavior of each technique.
- All tasks are solved, so the number of expansions is used as the metric for the quality of the learned heuristic.
    - Fewer expansions means better heuristics!

What is the contribution of the sampling algorithm?

Sampling method	BFS_{200}	DFS_{200}	RW_{200}	FSM_{200}
Expanded states	446.72	326.66	79.78	73.12

- BFS and DFS have extreme sample distributions. RW and our approach (FSM) have a balanced sample distribution over the state space.
- Blocks expands 5047 states with BFS and Grid 4102 with DFS, more than 25 and 10 times (resp.) than with the other techniques.

What is the contribution of the sampling algorithm?

Sampling method	FSM_{200}
Expanded states	73.12

What is the contribution of the sampling algorithm?

Sampling method	FSM_{200}
Expanded states	73.12

What is the contribution of the sampling algorithm?

Sampling method	FSM_{200}
Expanded states	73.12

What is the contribution of the sampling algorithm?

Sampling method	FSM_{200}
Expanded states	73.12

What is the impact of the regression limit?

Sampling method	FSM_{200}	FSM_{F}	$\mathrm{FSM}_{\bar{F}}$
Expanded states	73.12	63.36	69.48

What is the impact of the regression limit?

Sampling method	FSM_{200}	FSM_F	$FSM_{\bar{F}}$
Expanded states	73.12	63.36	69.48

- Our approaches outperform the baseline. F has the best performance, but...
- $\bar{F}$ outperforms the other techniques in 4 of the 7 domains.
  - Blocks expands 185 states, more than twice as many as the other regression limits.

How much do random samples improve the sample set?

	Expanded states	
Domain	0%	20%
Blocks	177.88	**57.00**
Grid	124.89	**66.52**
N-Puzzle	89.47	**80.93**
Rovers	17.03	**13.45**
Scanalyzer	55.29	**28.34**
Transport	**22.90**	25.95
VisitAll	30.90	**21.78**
Geo. mean	53.91	35.13

- Improves performance up to 20% of random samples, then degrades.
- Up to 50% of random samples the expanded states holds close (35.13 vs 38.76).

How much do our h-value improvement techniques refine the estimators?

	Mean difference h^*-h	
Domain	Baseline	Our approach
Blocks	24.01	0.18
Grid	13.60	0.61
N-Puzzle	70.87	5.11
Rovers	19.92	4.88
Scanalyzer	81.35	1.89
Transport	79.06	2.44
VisitAll	15.80	2.15
Geo. mean	33.45	1.60

- All our techniques (FSM, $\bar{F}$, SAI and SUI) improves by 20 times the approximation of the sample set estimates to h^*.
- Only $\bar{F}$ improves the mean difference to 5.56. Only SAI and SUI improves to 10.95.

What is the quality of the h-values produced by the learned heuristics?

Domain	Mean difference h^*-h h^{FF}	$\hat{h}_0$
Blocks	6.76	26.46
Grid	3.72	26.85
N-Puzzle	4.19	79.84
Rovers	0.17	11.08
Scanalyzer	2.78	106.37
Transport	1.13	109.77
VisitAll	1.31	21.55
Geo. mean	1.84	39.80

- The mean difference h^*-h over the forward state space of baseline $\hat{h}_0$ is more than 20 times greater than h^{FF}.

What is the quality of the h-values produced by the learned heuristics?

Domain	Mean difference h^*-h h^{FF}	$\hat{h}_0$	$\hat{h}_{\bar{F}}$
Blocks	6.76	26.46	2.91
Grid	3.72	26.85	2.73
N-Puzzle	4.19	79.84	6.75
Rovers	0.17	11.08	2.98
Scanalyzer	2.78	106.37	2.99
Transport	1.13	109.77	7.05
VisitAll	1.31	21.55	2.21
Geo. mean	1.84	39.80	3.57

- The mean difference h^*-h over the forward state space of baseline $\hat{h}_0$ is more than 20 times greater than h^{FF}.
- Our approach $\hat{h}_{\bar{F}}$ improves considerably: close to h^{FF}.

What is the quality of the h-values produced by the learned heuristics?

	Mean difference h^*-h			
Domain	h^{FF}	$\hat{h}_0$	$\hat{h}_{\bar{F}}$	$\hat{h}_{\bar{F}}^{20\%}$
Blocks	6.76	26.46	2.91	2.42
Grid	3.72	26.85	2.73	9.78
N-Puzzle	4.19	79.84	6.75	12.73
Rovers	0.17	11.08	2.98	6.35
Scanalyzer	2.78	106.37	2.99	9.01
Transport	1.13	109.77	7.05	14.89
VisitAll	1.31	21.55	2.21	4.74
Geo. mean	1.84	39.80	3.57	7.40

- The mean difference h^*-h over the forward state space of baseline $\hat{h}_0$ is more than 20 times greater than h^{FF}.
- Our approach $\hat{h}_{\bar{F}}$ improves considerably: close to h^{FF}.
- Adding random samples worsens the mean difference h^*-h.

Does the quality of the h-values reflect in the search?

Heuristic	h^*
Expanded states	14.53

Does the quality of the h-values reflect in the search?

Heuristic	h^*	h^{FF}
Expanded states	14.53	38.98

Does the quality of the h-values reflect in the search?

Heuristic	h^*	h^{FF}	$\hat{h}_0$
Expanded states	14.53	38.98	81.86

Does the quality of the h-values reflect in the search?

Heuristic	h^*	h^{FF}	$\hat{h}_0$	$\hat{h}_{\bar{F}}$
Expanded states	14.53	38.98	81.86	53.91

- Remains the relative order of quality of estimates over the forward state space.

Does the quality of the h-values reflect in the search?

Heuristic	h^*	h^{FF}	$\hat{h}_0$	$\hat{h}_{\bar{F}}$	$\hat{h}_{\bar{F}}^{20\%}$
Expanded states	14.53	38.98	81.86	53.91	35.13

- Remains the relative order of quality of estimates over the forward state space.
- But not. Random samples worsen the quality of estimates but reduce the expanded states.
- Our approach with random samples ($\hat{h}_{\bar{F}}^{20\%}$) outperforms h^{FF}.

Large State Spaces

- Validate our findings from experiments in small state spaces...
  - ...and how does our approach compare with logic-based heuristics and other learned heuristics in the literature?
- Coverage (percentage of solved tasks) is used as the metric for the quality of the learned heuristic.
  - Greater coverage means better heuristics!

Comparison of Heuristic Functions

Domain	Coverage (%) h^{FF}	h^{GC}	$\hat{h}_0$	$\hat{h}^{20\%}_{\bar{F}}$
Blocks	100.00	100.00	100.00	100.00
Depot	94.33	80.00	57.19	89.26
Grid	94.00	51.00	38.11	60.33
N-Puzzle	92.50	4.00	13.75	86.81
Pipes-NT	63.40	89.40	13.51	79.84
Rovers	85.50	66.00	13.53	15.39
Scanalyzer	100.00	100.00	59.70	73.67
Storage	33.00	13.50	1.94	27.67
Transport	100.00	100.00	48.89	100.00
VisitAll	92.00	100.00	74.19	98.85
Mean	85.47	70.39	42.08	73.18

- h^{FF} dominates in most domains.
- Our approach $\hat{h}^{20\%}_{\bar{F}}$ improves the baseline $\hat{h}_0$ by about 31%, with competitive coverage in most domains.
- Also, outperforms h^{GC} (73.18 vs 70.39) with higher or equal coverage in 6 out of 10 domains.

Comparison to Other Methods

Technique	Avg. coverage
$\hat{h}^{\text{Boot}}$ (Ferber *et al.*, 2022)	45.40
$\hat{h}^{\text{N-RSL}}$ (O'Toole *et al.*, 2022)	58.80
$\hat{h}^{20\%}_{\bar{F}}$ (Our approach)	72.82

- All techniques use the same test dataset and NN architecture.
- $\hat{h}^{\text{Boot}}$ uses considerably more computational resources than the other techniques.
- Our approach outperforms other techniques.

Conclusion

Conclusion

- A distribution covering various portions of the state space without repeated samples close to the goal works best.
- Both the sample distribution and reasonable cost-to-goal estimates contribute to search performance.
- The *h*-value improvement technique SUI and adaptative rollout limit $\bar{F}$ have the most positive impact on sampling quality.
- Our approach outperforms h^{GC} but not h^{FF}.
  - h^{FF} has access to the entire model, whereas our approach only uses a minimal part of it.
- Approaches with few samples and computational resources can also be efficient.

Questions?

rvbettker@inf.ufrgs.br

References

Ferber, Patrick, Geißer, Florian, Trevizan, Felipe, Helmert, Malte, & Hoffmann, Jorg. 2022. Neural Network Heuristic Functions for Classical Planning: Bootstrapping and Comparison to Other Methods. *In: Proceedings of the Thirty-Second International Conference on Automated Planning and Scheduling*. Online: AAAI Press.

O'Toole, Stefan, Ramirez, Miquel, Lipovetzky, Nir, & Pearce, Adrian R. 2022. Sampling from Pre-Images to Learn Heuristic Functions for Classical Planning. *Pages 308–310 of: Fifteenth International Symposium on Combinatorial Search*, vol. 15. Vienna, Austria: AAAI Press.

Shen, William, Trevizan, Felipe, & Thiebaux, Sylvie. 2020. Learning Domain-Independent Planning Heuristics with Hypergraph Networks. *Pages 574–584 of: Proceedings of the Thirty International Conference on Automated Planning and Scheduling*. Online: AAAI Press.

Yu, Liu, Kuroiwa, Ryo, & Fukunaga, Alex. 2020. Learning Search-Space Specific Heuristics Using Neural Network. *In: Proceedings of the 12th Workshop on Heuristics and Search for Domain-independent Planning*. Online: ICAPS.