Bounded-Suboptimal Search with Learned Heuristics

Matias Greco[1] Jorge A. Baier[1,2]
[1]Pontificia Universidad Católica de Chile
[2]Instituto Milenio Fundamentos de los Datos

Abstract

Reinforcement learning allows learning very accurate heuristics for hard combinatorial puzzles like the 15-puzzle, the 24-puzzle, and Rubik's cube. In this paper, we empirically investigate how to exploit these learned heuristics in the context of (deterministic) heuristic search with bounded suboptimality guarantees, using the learned heuristic for the 15 and 24-puzzle of DeepCubeA. We show that Focal Search (FS), in its most straightforward form, that is, using the learned heuristic to sort the focal list, has poor performance when compared to ***Focal Discrepancy Search (FDS)****, a version of FS that we propose that uses a discrepancy function to sort the focal list. This is interesting the best performing algorithm \emph{does not use} the heuristic values themselves but just the ranking between the successors of the node. In addition, we show FDS is competitive with satisficing search algorithms Weighted A* and Greedy Best-First Search.*

Introduction

Recent work has shown that reinforcement learning can learn very accurate heuristics estimators in different scenarios, such as domain-independent automated planning (Ferber, Helmert, and Hoffmann 2020) and combinatorial puzzles such as Rubik's cube and the sliding tile puzzle (Agostinelli et al. 2019).

Given an accurate learned heuristic function, a natural question to ask is *how to exploit such a function within a bounded-suboptimal search algorithm*; i.e, an algorithm that provides guarantees on the returned solution. This question is challenging because learned heuristics, even if highly accurate, cannot be assumed be admissible.

This paper study how to exploit a learned heuristic within a bounded-suboptimal search algorithm, that also uses an admissible heuristic to provide suboptimality bounds. Following the previous work of Araneda, Greco & Baier (2021), which studied various ways in that Focal Search (FS) can be integrated with a learned policy; we explore how FS can be used with a highly effective learned heuristic, such as DeepCubeA.

Learned Policies and Learned Heuristics

A Learned Policy is a function π that maps each state-action pair to a probability distribution over the actions:

$$\pi : A, S \rightarrow [0, 1]$$

Therefore π(a,s) is such that $\sum_{a \in A} \pi(a, s) = 1$ for every state s

A Learned Heuristic is a function h_{nn} which maps a state s to a prediction of its cost-to-go.

$$h_{nn} : S \rightarrow \mathbb{R}^{0,+}$$

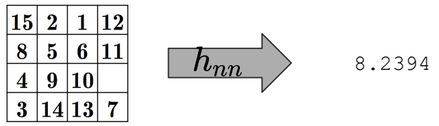

Figure 1: Operation of a Learned policy or Learned heuristic

Focal Search (FS)

FS is a well-known Bounded-Suboptimal Search algorithm that uses inadmissible heuristic functions, which may not be cost-to-go estimates. FS uses two lists: `OPEN`, which is the search frontier sorted in ascending order by f=g+h, and `FOCAL`, which is sorted arbitrarily.

This algorithm chooses for expansion the best node (in terms of h_{FOCAL}) in focal.

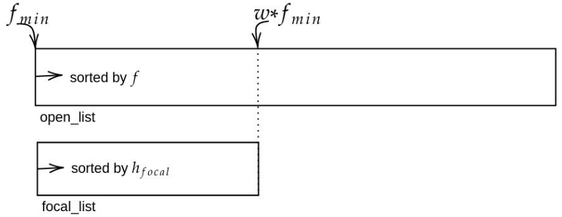

Figure 2: Focal Search

Focal Discrepancy Search

According to the original definition (Harvey & Ginsberg, 1995), a discrepancy occurs over a path when at a certain state of the path s, the action taken does not lead to child of s with minimum *h-value*.

We define a preferred action at state s as the node with the most promising heuristic value between the successors

$$pref\_state(s) = \text{argmin}_{s' \in succ(s)} \{h(s')\}$$

Assuming that s is a state which during state has been reached via path $\sigma_s = s_1 s_2 \dots s_n$ (with $s_1 = s_{start}$ and $s_n = s$), we define $N_{\text{nonpref}}(\sigma_s)$ as the number of times along the path in which the state with best heuristic value was not taken. Thus, we define a discrepancy as:

$$h_{\text{disc\_best}}(s) = N_{\text{nonpref}}(\sigma_s)$$

Discrepancies were originally proposed for binary trees. We also consider counting discrepancies according to their successor rank (Karoui et al. 2007)

$$h_{\text{disc\_rank}}(s) = \text{rank}(\sigma_s)$$

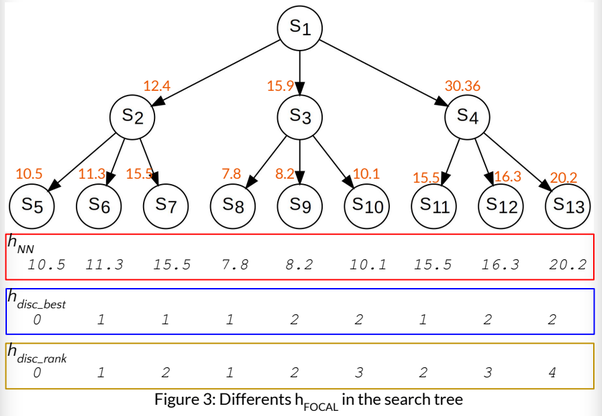

Figure 3: Differents h_{FOCAL} in the search tree

Results

We use the pre-trained models of DeepCubeA (Agostinelli et al. 2019) for the 15- and 24-puzzle as learned heuristic. We test over the Korf's instances. We use our approach against other bounded suboptimal algorithms such as wA* and Focal Search using the learned heuristic as h_{FOCAL}.

Table 1: Results for the 15-puzzle with the algorithms using a suboptimality bound $w = 1.5$

	Coverage	Expansions (avg)	Cost (avg)
*wA**	100%	22100	56.67
FS(h_nn)	83%	10414	54.57
FDS(best)	100%	1478	55.47
FDS(rank)	100%	6542	55.45

Table 2: Results for the 24-puzzle with the algorithms using a suboptimality bound $w = 1.5$

	Coverage	Expansions (avg)	Cost (avg)
*wA**	68%	1519344	112.20
FS(h_nn)	96%	4465	111.5
FDS(best)	100%	137	110.26
FDS(rank)	100%	139	109.98

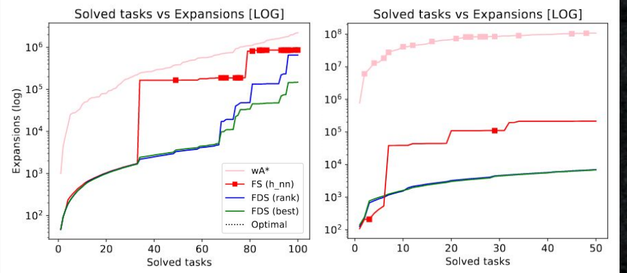

Figure 4: Results in 15- and 24-Puzzle (resp.) against bounded suboptimal algorithm wA* for a set w = 1.5

We also compare FDS against satisficing algorithms, such as Greedy-BFS, wA* using the learned heuristic, and prefA*. The results show that it is still competitive and has the advantage of returning suboptimality bounds.

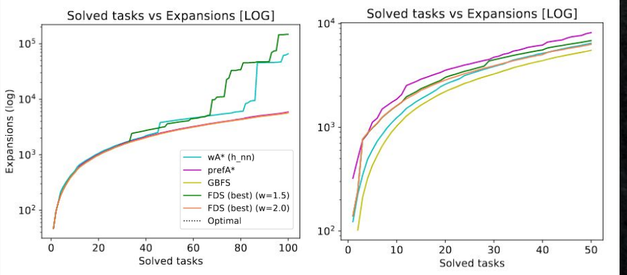

Figure 5: Results in 15- and 24-Puzzle (resp.) of satisficing algorithm (wA*hnn, prefA*, GBFS) against bounded suboptimal FDS for a set w = 1.5 and 2.0

REFERENCES
- Araneda, P. Greco, M. Baier, J. A. (2021). Exploiting Learned Policies in Focal Search. In SoCS (pp. 2-10).
- Agostinelli, F., McAleer, S., Shmakov, A., & Baldi, P. (2019). Solving the Rubik's cube with deep reinforcement learning and search. *Nature Machine Intelligence*,
- Harvey, W. D., & Ginsberg, M. L. (1995, August). Limited discrepancy search.

2th ICAPS Workshop on Bridging the Gap Between AI Planning and Reinforcement Learning - 2021

Supported by National Agency for Research and Development (ANID), Chile / Doctorado Nacional / 2019 - 21192036