Proceedings of the Fifteenth International Symposium on Combinatorial Search (SoCS 2022)

Avoiding Errors in Learned Heuristics in Bounded-Suboptimal Search

Matias Greco[1], **Jorge A. Baier**[1,2]

[1] Departamento de Ciencia de la Computación, Pontificia Universidad Católica de Chile
[2] Instituto Milenio Fundamentos de los Datos, Chile
mogreco@uc.cl, jabaier@ing.puc.cl

Abstract

Despite being very effective, learned heuristics in bounded-suboptimal search can produce heuristic plateaus or move the search to zones of the state space that do not lead to a solution. In addition, it produces inadmissible cost-to-go estimates; therefore, it cannot be exploited with classical algorithms like WA* to produce w-optimal solutions. In this paper, we present two ways in which Focal Search can be modified to exploit a learned heuristic in a bounded suboptimal search: Focal Discrepancy Search, which, to evaluate each state, uses a discrepancy score based on the best-predicted heuristic value; and K-Focal Search, which expands more than one node from the FOCAL list in each expansion cycle. Both algorithms return w-optimal solutions and explore different zones of the state space than the ones that focal search, using the learned heuristic to sort the FOCAL list, would explore.

Introduction

Many machine learning algorithms have demonstrated the capacity to learn very effective heuristics estimates to guide search in a variety of problems, such as domain-independent planning (Shen, Trevizan, and Thiébaux 2020), Sokoban (Groshev et al. 2018), Rubik's cube, and the sliding-tile puzzles (Agostinelli et al. 2019). The learned heuristics produce inadmissible cost-to-go estimates, making it impossible to exploit them in algorithms that support its suboptimality guarantees on admissible heuristics. On the other hand, despite being very effective, it can make mistakes that guide the search to uninformed heuristics regions or heuristics plateaus, which degrade the search efficiency. For that reason, the question of how to exploit a learned heuristic in a bounded-suboptimality search is still an open question. DeepCubeA is a very effective learned heuristic to solve puzzle problems, such as sliding-tile puzzles or the Rubik's cube, trained with reinforcement learning. Despite that being very effective, the predicted cost-to-go can have large errors. Figure 1 shows the difference of the DeepCubeA (Agostinelli et al. 2019) learned heuristic (h_{nn}) with respect to the optimal cost (h^*). It is observed that the further away the state is from the goal, the greater its average error.

To avoid concentrating the search in zones of the search space where the learned heuristic may be uninformed, which guide the search to zones that do not lead to a solution, we propose to combine it with a calculated admissible heuristic in Focal Search (Pearl and Kim 1982), which can help to not concentrate the search only in these zones.

In this paper, we present two variations of the well-known bounded suboptimality search algorithm Focal Search: Focal Discrepancy Search (FDS), which exploits the discrepancy score (i.e., the numbers of nodes in the path in which the action with the best heuristic value was not taken); and K-Focal Search (K-FS) which, instead of select for expansion the best node at expansion cycle, select the best K nodes, expanding nodes that may not be expanded in a regular best-first order. Both algorithms keep the theoretical guarantees of Focal Search, i.e., return w-optimal solutions.

On the experimental side, we evaluate the algorithms on the 24-puzzle benchmark using DeepCubeA. We compare our algorithms against Focal Search using the learned heuristic to sort the `FOCAL`, similar was proposed by Spies et al., and two purely heuristic search algorithms that exploit the admissibility of the heuristic, such as WA* and DPS. Our results show that by using a learned heuristic, our algorithms outperform the classical Focal Search by two orders of magnitude and WA* and DPS by four orders of magnitude regarding the number of expansions.

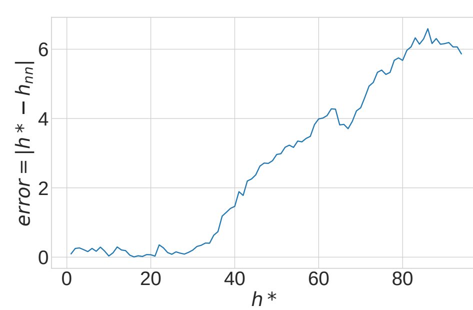

Figure 1: Error of the DeepCubeA learned heuristic (h_{nn}) on 24-puzzle with respect to $h*$

Focal Discrepancy Search

Focal Discrepancy Search (FDS) (Araneda, Greco, and Baier 2021; Greco, Araneda, and Baier 2022) is a version of Focal Search which sorts `FOCAL` by the discrepancy associated with the path of each state. More formally, if s is a state in `FOCAL`, at any point during the execution of FS, its priority is given by $disc(path(s))$, which is the number of times along the path where the state with the best heuristic value was not selected for expansion. This method was originally proposed to use a learned policy, but it is also applicable to learned heuristics, even in an anytime context.

K-Focal Search

K-Focal Search (K-FS(k)) is a generalization of Focal Search. Instead of selecting for expansion the best node in `FOCAL`, K-FS(k) extracts the best k nodes from `FOCAL`, and unless the goal is among the extracted states, it expands all such states. If the `FOCAL` list contains less than k nodes, it selects for expansion all nodes in the `FOCAL` list. K-FS with $k = 1$ [K-FS(1)] executes the exact procedure that Focal Search. Due to all states in `FOCAL` are within the bound, K-FS keeps the theoretical guarantees of Focal Search, thus returning a w-optimal solution.

K-FS was mainly proposed to reduce the time expended in calculating the learned heuristic for each state, accumulating a batch of states whose heuristics estimates will be calculated using a GPU via batched computation. K-FS can perform a more efficient search because it includes more exploration and expand nodes that might not have been expanded in a standard best-first search algorithm.

Experimental Results

We evaluated our algorithm on the 24-puzzle with a suboptimality bound $w = 1.5$. We use the trained models of DeepCubeA (Agostinelli et al. 2019) as learned heuristic. This model was implemented in Pytorch and is publicly available. We use the Linear Conflict heuristic as an admissible heuristic. All algorithms were implemented in Python 3, and the experiments were run on an Intel Xeon E5-2630 machine with 64GB RAM, using a single CPU core and one GPU Nvidia Quadro RTX 5000. For all experiments, we use a 30-minute timeout. As evaluations, we use Korf's 50 instances for the 24-puzzle (Korf and Felner 2002). Our algorithms are compared with FS (*FS*) using the same neural-net heuristic, which calculates the learned heuristic for each node that is inserted in `FOCAL` at the moment that it is inserted; and two other state-of-the-art bounded suboptimality search algorithms: Weighted A* (*WA**) and Dynamic Potential Search (*DPS*) (Gilon, Felner, and Stern 2016), which is a bounded-suboptimal version of potential search.

The results show that purely heuristic search algorithms, such as WA* and DPS, can solve only 68% and 80% of problems, resp, FS and K-FS(1) can solve 96% of instances; and FDS and K-FS(5) and K-FS(10) solve 100% of the instances. Regarding the number of expansions, we observe that FDS and K-FS(5) perform two orders of magnitude fewer expansions than FS and K-FS(1) and four orders of magnitude than WA* and DPS. Regarding the time spent in the search, we observe that K-FS(5) and K-FS(10) perform slightly more expansions than FDS, and spend less time due to the GPU acceleration. Note that FS and K-FS(1) select nodes for expansion in the same order, but due to batched computation, K-FS can perform more expansion spending the same time. Regarding the best quality of solution, we observe that the best solutions were obtained by K-FS(10).

	w=1.5			
	Cov.	Exp.	Cost	Time
WA*	68	2362299	112.21	1015.86
DPS	80	1753895	113.83	785.0
FS	96	31689	111.50	84.15
FDS	100	340	110.26	1.40
K-FS(1)	96	43464	111.50	80.37
K-FS(5)	100	523	103.64	0.81
K-FS(10)	100	1017	102.24	1.02

Table 1: Results on 24-puzzle

Conclusions and Future Work

In this paper, we present two algorithms based on Focal Search applicable to avoid large errors or plateaus on learned heuristics in bounded suboptimal search, keeping the theoretical guarantees provided by Focal Search. On the experimental side, we demonstrate the effectiveness of our algorithms in the 24-puzzle domain using DeepCubeA, a very effective inadmissible learned heuristic. We show that our approach outperforms other bounded-suboptimal heuristic search algorithms such as WA* and DPS by four orders of magnitude and FS using the learned heuristic by two orders of magnitude regarding the number of expansions. We hypothesize that the success of our algorithms is because they can explore different zones of the state space due to the diversity introduced by its expansion strategy in the case of K-FS, or its evaluation function in the case of FDS. In future work, we want to combine both algorithms, i.e., sort `FOCAL` by the discrepancy (as FDS does), expanding the best K nodes from `FOCAL`, which has the best discrepancy score (as K-FS does). Another exciting line of research is to include a random type exploration as type-WA* does (Cohen, Valenzano, and McIlraith 2021) and incorporate our expansion strategy in the recent versions of EES (Fickert, Gu, and Ruml 2022).

Acknowledgements

Matias Greco was supported by the National Agency for Research and Development (ANID) / Doctorado Nacional / 2019 - 21192036. We also thank to Centro Nacional de Inteligencia Artificial CENIA, FB210017, BASAL, ANID, for partially funding the authors.

References

Agostinelli, F.; McAleer, S.; Shmakov, A.; and Baldi, P. 2019. Solving the Rubik's cube with deep reinforcement learning and search. *Nature Machine Intelligence*, 1(8): 356–363.

Araneda, P.; Greco, M.; and Baier, J. A. 2021. Exploiting Learned Policies in Focal Search. In Ma, H.; and Serina, I., eds., *Proceedings of the Fourteenth International Symposium on Combinatorial Search, SOCS 2021, Virtual Conference [Jinan, China], July 26-30, 2021*, 2–10. AAAI Press.

Cohen, E.; Valenzano, R. A.; and McIlraith, S. A. 2021. Type-WA*: Using Exploration in Bounded Suboptimal Planning. In Zhou, Z., ed., *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, 4047–4053. ijcai.org.

Fickert, M.; Gu, T.; and Ruml, W. 2022. New Results in Bounded-Suboptimal Search. In *Thirty-Six AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Four Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelfth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022, Virtual Event, February 22 - March 1, 2022*. AAAI Press. To appear.

Gilon, D.; Felner, A.; and Stern, R. 2016. Dynamic Potential Search - A New Bounded Suboptimal Search. In Baier, J. A.; and Botea, A., eds., *Proceedings of the Ninth Annual Symposium on Combinatorial Search, SOCS 2016, Tarrytown, NY, USA, July 6-8, 2016*, 36–44. AAAI Press.

Greco, M.; Araneda, P.; and Baier, J. 2022. Focal Discrepancy Search for Learned Heuristics. In *Proceedings of the Fifteenth International Symposium on Combinatorial Search, SOCS 2022, Vienna, Austria, July 21-23, 2022*. To appear.

Groshev, E.; Goldstein, M.; Tamar, A.; Srivastava, S.; and Abbeel, P. 2018. Learning Generalized Reactive Policies Using Deep Neural Networks. In de Weerdt, M.; Koenig, S.; Röger, G.; and Spaan, M. T. J., eds., *ICAPS*, 408–416. AAAI Press.

Korf, R. E.; and Felner, A. 2002. Disjoint pattern database heuristics. *Artificial intelligence*, 134(1-2): 9–22.

Pearl, J.; and Kim, J. H. 1982. Studies in Semi-Admissible Heuristics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(4): 392–399.

Shen, W.; Trevizan, F. W.; and Thiébaux, S. 2020. Learning Domain-Independent Planning Heuristics with Hypergraph Networks. In *ICAPS 2020*, 574–584. AAAI Press.

Spies, M.; Todescato, M.; Becker, H.; Kesper, P.; Waniek, N.; and Guo, M. 2019. Bounded Suboptimal Search with Learned Heuristics for Multi-Agent Systems. In *AAAI 2019*, 2387–2394. AAAI Press.