Learning to Estimate Search Progress Using Sequence of States

TECHNION Israel Institute of Technology

Matan Sudry and Erez Karpas, Faculty of Industrial Engineering and Management, Technion, Israel

Motivation

Suppose we have a temporal problem and we need to find a solution using search.

Search progress estimation can be used to make decisions about search.

Estimates of remaining search time have been used in situated temporal planning [1] and in Branch-and-Bound search [2].

Current search progress estimators [3], which were developed manually using extensive expertise in heuristic search and human creativity.

Problem Statement

many problems of interest can be modeled as state-space search problems. A state space is formally represented as a tuple $T = \langle S, S_0, S_G, A, f, c \rangle$, where:

- S is a finite and non-empty set of states,
- $S_0 \in S$ is the initial state,
- $S_G \subseteq S$ is a non-empty set of goal states,
- $A(s) \subseteq A$ denotes the actions applicable in each state $s \in S$,
- $f : S \times A \to S$ is the state transition function
- $c(s, a)$ is the cost of performing action a in state s

The search progress of algorithm A solving problem P after expanding $\mathrm{Gen}_A(P)$ nodes is: $\mathrm{Prog}_A(\mathrm{E}_A(P)) = \frac{\mathrm{Gen}_A(P)}{\mathrm{Gen}_A(P)+\mathrm{Rem}_A(P,\mathrm{Gen}_A(P))}$

Current methods

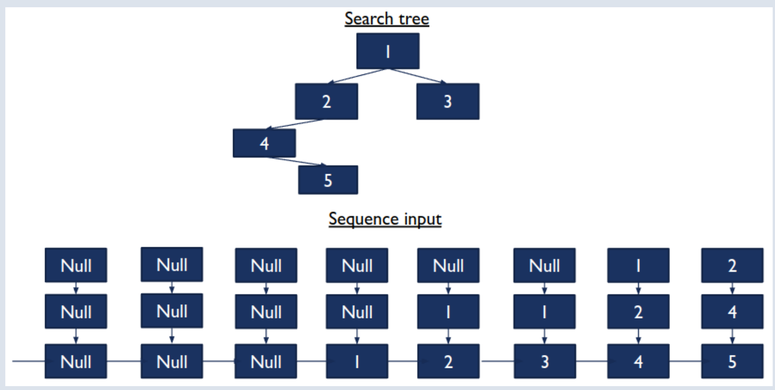
Velocity-Based Search Speed Estimator (VeSP): $V = \frac{h_0 - h_{\min}}{\mathrm{Gen}_A(P)}$, $SE_V = \frac{h_{\min}}{V}$, $VeSP(\mathrm{Gen}_A(P)) = \frac{\mathrm{Gen}_A(P)}{\mathrm{Gen}_A(P)+SE_V}$

Vacillation-Based Search Speed Estimator (VaSP): $SE_e = \overline{\Delta e} \cdot h_{\min}$, $VaSP(\mathrm{Gen}_A(P)) = \frac{\mathrm{Gen}_A(P)}{\mathrm{Gen}_A(P)+SE_e}$

Path-Based Progress Estimator (PBP): $PBP(Exp) = \frac{g(n)}{g(n)+h(n)}$

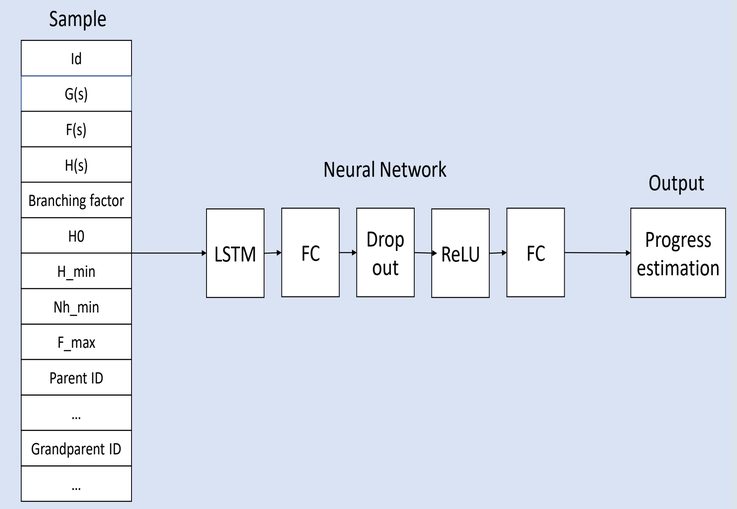
Distribution-Based Progress Estimator (DBP): $Prog^*(\mathrm{Gen}_A(P)) = \frac{\mathrm{Gen}_A(P)}{\sum_{i=1}^{m} \hat{c}[d_i]}$

Search tree to sequence input

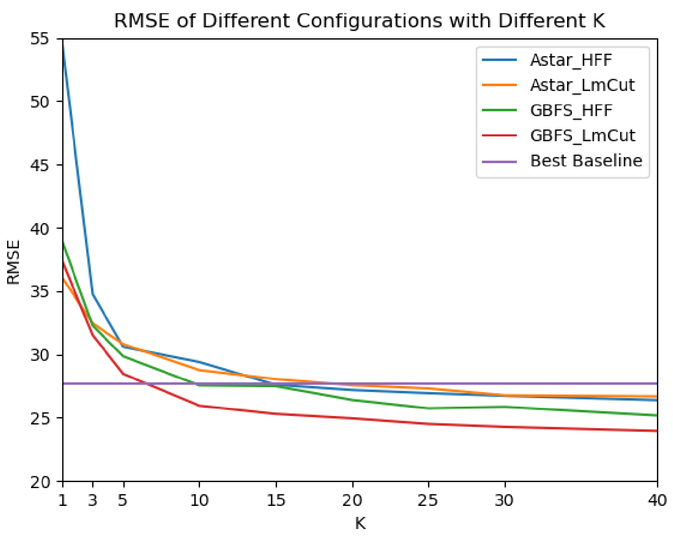


Search tree convert to a sequence input

Our solution



K experiment



RMSE with Different K and Different Configurations

Training regimes

Other Domains (OD): In this regime, when we test on problems from some domain, we train on all problems from all *other* domains.

Same Domain (SD): In this regime, when we test on problems from some domain, we train only on problems from the same domain.

Other Domains Tune Same (ODTS): In this regime, we first train a predictor based on instances from the other domains. We then take this trained model, and fine tune it on one half of the instances from the target domain (split, as before, to even and odd numbered problems).

Results

	A/L	A/H	G/L	G/H
S	121	119	89	204
B	9/19	5/18	11/15	15/20
M	0.82	0.94	0.71	0.73

S=Solved problems, B=Best accuracy, M=RMSE ours/best predictor. A=A*, G=GBFS, H=h_{FF} and L=Lmcut.

Search	heuristic	OD	SD	ODTS
A	H	1	1	**5**
A	L	1	1	**5**
G	H	5	2	**6**
G	L	2	1	**4**

The numbers indicate the number of domains this regime has the best results, A=A*, G=GBFS, H=h_{FF} and L=Lmcut

Generalization

Test/Train	G/L	A/L	G/H	A/H
G/L	**24.0**	28.3	28.3	28.2
A/L	**26.1**	26.7	27.2	27.6
G/H	25.7	29.3	**25.2**	26.8
A/H	27.1	29.2	26.8	**26.4**

RMSE (in percent) on generalization evaluation. results in bold are the best performance in this row. A=A*, G=GBFS, H=h_{FF} and L=Lmcut

Conclusions

We presented a novel approach to search progress estimation using deep learning.

On IPC domains, our approach achieve better performance of 6-29 percent from current methods

We presented three different regimes of three different applications

Our algorithm generalize between different search algorithm and heuristic

References

[1] M. Cashmore, A. Coles, B. Cserna, E. Karpas, D. Magazzeni, and W. Ruml, "Temporal planning while the clock ticks," 2018.

[2] D. Anderson, G. Hendel, P. Le Bodic, and M. Viernickel, "Clairvoyant restarts in branch-and-bound search using online tree-size estimation.," in *AAAI*, 2019.

[3] J. T. Thayer, R. Stern, and L. H. Lelis, "Are we there yet?-estimating search progress.," in *SOCS*, 2012.