Forward Backward Novelty Search

[1] **Anubhav Singh,** [1] **Chao Lei,** [1] **Nir Lipovetzky,** [2] **Miquel Ramirez,** [3] **Javier Segovia-Aguas**
[1] School of Computing and Information Systems, University of Melbourne, Australia
[2] Electrical and Electronic Engineering, University of Melbourne, Australia
[3] Dept. Information and Communication Technologies, Universitat Pompeu Fabra, Spain
{anubhavs, clei1}@student.unimelb.edu.au, {nir.lipovetzky, miquel.ramirez}@unimelb.edu.au, javier.segovia@upf.edu

Abstract

It has been shown recently that width-based search algorithms can be employed to search over the regression space (backward search). While many of the benchmarks are challenging for the width-based backward search, it performs significantly better than the forward counterparts on certain domains. This *orthogonal* behavior of forward and backward width-based search is quite suitable for an integrated approach. Indeed, it has been shown that a simple forward-backward integration which runs forward best-first width search (BFWS) with novelty pruning followed by the backward counterpart results in better coverage than both. Similarly, pairing forward-backward *pruned* BFWS algorithm with the state-of-the-art *Dual-BFWS* improves the overall coverage over the IPC *satisficing* benchmark. In this paper, we present an integration of approximate novelty search with the forward-backward BFWS.

Backward Best-First Width Search

Lei and Lipovetzky (2021) showed that BFWS and k-BFWS (Lipovetzky and Geffner 2017a,b) can be adapted to solve the regression state model directly. The definition of novelty is the same in both directions, as it only depends on the syntax of the states, i.e. the state variables. The critical paths heuristic h^2 (Haslum and Geffner 2000; Alcázar and Torralba 2015) is generated from s_0, the initial state of the forward model, to extract the set of forward mutex fluent pairs (Blum and Furst 1997). Mutexes are used to prune partial states in the regression unreachable from s_0, and hence a generated state s is pruned if it contains a mutex pair $h^2(p, q) = \infty$, $p, q \in s$. The goal counter instead of keeping track of the number of unrealized forward goals $g \in G$ in progression, $\#g(s) = |s \cap I| + |s \setminus I|$ keeps track of the number of initial state fluents I achieved, as well as the number of fluents that still have to be removed from s to reach one of the regression goal states. The goal counter is further strengthened by creating an *I-ordering* $p < q$ of fluents $p, q \in I$ when all actions requiring p *edelete* q. In regression, an action a *edeletes* a fluent q if $q \in add(a)$ or $\exists_{p \in pre(a) \cup del(a)} h^2(p, q) = \infty$. This I-ordering graph refines the goal counter $\#g(s)$ by counting as achieved fluents $p \in s, p \in I$ whose precedences are satisfied in s.

Width-Based Forward Backward Search

The experiment results introduced by (Lei and Lipovetzky 2021) show that forward and backward search can be orthogonal. One of prominent combinations of forward search and backward search is *FB* where F stands for forward k-BFWS(f_5) (Lipovetzky and Geffner 2017b), B for the backward counterpart, and $k = 2$. *FB* is a simple integration where F is run first and then B runs only if F stops with no solution. *FB* solves the most instances, 794 over 1095 test instances from 42 domains introduced in the satisficing tracks of IPCs 1998–2018, 60 problems more than F over 10 different domains. This is witnessed further by the results over *Dual-BFWS*, runner-up on the last satisficing track at the IPC-2018 (Francès et al. 2018). *Dual-BFWS* runs first a forward F with $k = 1$, and a second complete BFWS (Lipovetzky and Geffner 2017a) if no solution is found. The results show that running *FB* with $k = 1$ first instead improves the state-of-the-art (*Dual-FB*). *FB* with $k = 1$ can be thought of as a quick preprocessing step that could be integrated in every state-of-the-art planner as it either solves a problem or fails fast.

Forward Backward Sequential BFWS(f_5) planner

In Forward Backward Sequential BFWS(f_5), we iteratively call forward and backward variant of BFWS(f_5) until we find a solution or run out of time. The algorithm *approximates* the *state novelty* and uses an *adaptive policy* to control the open-list (Singh et al. 2021) which allows the planner to give space and time guarantees on the function that computes the novelty measure. The planner makes recursively calls to forward and backward k-BFWS(f_5), increasing k by 1 at each iteration, i.e. it begins by calling forward k-BFWS(f_5), $k = 1$, in which nodes of novelty greater than 1 are pruned. If the search fails to find a solution, then it calls the backward counterpart. If it fails again, then it repeats the steps with $k = 2$, and so on. As an optimization step, based on *empirical* reasoning, we stopped the backward runs for $k \geq 2$ in the planner that we have submitted into the IPC. We have submitted the planner for both *agile* and *satisficing* track with one difference - after finding a plan using Forward Backward Sequential BFWS, the *satisficing* variant optimizes the plans with weighted A* used in

LAMA (Richter and Westphal 2010) until timeout.

References

Alcázar, V.; and Torralba, A. 2015. A Reminder about the Importance of Computing and Exploiting Invariants in Planning. In *Int'l Conference on Automated Planning and Scheduling (ICAPS)*, ICAPS, 2–6.

Blum, A. L.; and Furst, M. L. 1997. Fast Planning through Planning Graph Analysis. *Artificial Intelligence Journal (AIJ)* 90: 281–300.

Francès, G.; Geffner, H.; Lipovetzky, N.; and Ramírez, M. 2018. Best-First Width Search in the IPC 2018: Complete, Simulated, and Polynomial Variants (planner abstract). *Proc. of the Int'l Planning Competition (IPC)* .

Haslum, P.; and Geffner, H. 2000. Admissible Heuristics for Optimal Planning. In *Int'l Conference on Automated Planning and Scheduling (ICAPS)*, AIPS, 140–149.

Lei, C.; and Lipovetzky, N. 2021. Width-based backward search. In *Int'l Conference on Automated Planning and Scheduling (ICAPS)*, volume 31, 219–224.

Lipovetzky, N.; and Geffner, H. 2017a. Best-First Width Search: Exploration and Exploitation in Classical Planning. In *Proc. of the AAAI Conference (AAAI)*.

Lipovetzky, N.; and Geffner, H. 2017b. A polynomial planning algorithm that beats LAMA and FF. In *Int'l Conference on Automated Planning and Scheduling (ICAPS)*.

Richter, S.; and Westphal, M. 2010. The LAMA planner: Guiding cost-based anytime planning with landmarks. *Journal of Artificial Intelligence Research (JAIR)* 39: 127–177.

Singh, A.; Lipovetzky, N.; Ramirez, M.; and Segovia-Aguas, J. 2021. Approximate novelty search. In *Int'l Conference on Automated Planning and Scheduling (ICAPS)*, volume 31, 349–357.