Approximate Novelty Search

[1] **Anubhav Singh,** [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@student.unimelb.edu.au, {nir.lipovetzky, miquel.ramirez}@unimelb.edu.au, javier.segovia@upf.edu

Abstract

Width-based search is an effective approach to classical planning which has produced many successful algorithms over the years. A key feature which distinguishes width-based search from classic heuristic search algorithms is the use of *specific structural properties* of the *explored state space* to guide the *exploration* and goal-directed heuristic measures for *exploitation*. The *structural properties* are captured as an n-ary relation over the *fluents* which is processed to compute the *state novelty*. The size of the relation and the time complexity of computing novelty measure is exponential on the arity n. Approximate novelty search introduces novel polynomial approximations of *state novelty* and width-based search. It uses *Bloom filter* to *efficiently* represent the interpretation of the relational predicate and *random sampling* in the computation of *state novelty*. It also uses an adaptive policy which decides to delay the generation of successor states. In this paper, we explain the integration of these two techniques into the *polynomial-time* variant of Best-First Width Search (BFWS), one of the most successful width-based algorithm in *satisficing* planning.

.

Introduction

Width-based search algorithms rely on the notion of *state novelty* which is an *orthogonal* measure to *goal-directed* heuristics. While the heuristics provide an approximation of the distance to the goal, the novelty measures instead capture how *novel* the state is with respect to the *explored* state space. Several width-based search algorithms have been proposed (Lipovetzky and Geffner 2014; Lipovetzky et al. 2014; Lipovetzky and Geffner 2017a,b; Francès et al. 2018; Katz et al. 2018) out of which best-first width search (BFWS) has been the most acclaimed. A major shortcoming of the width-based methods is that the complexity of computing novelty measure is exponential on the number of discrete level or categories used to rank the states. While there exists an upper bound on the number of novelty categories required to solve a given classical planning instance (Lipovetzky and Geffner 2012), a large bound results in impractical space and time requirements for novelty computation. Approximate Novelty Search (Singh et al. 2021) proposes a probabilistic approximation of novelty measure which trades off accuracy of novelty computation for commitments on space and time complexity. This allows width-based search algorithms to tap into the search space associated with higher novelty categories. Next, we present a brief account of best-first width search and novelty approximation, along with the description of the planner configurations that we have submitted in *agile* and *satisficing* track of the IPC.

Approximate Novelty Search

BFWS (Lipovetzky and Geffner 2017a) is a best-first search algorithm which uses a tuple of functions $f(n) = (w, h_1, \ldots, h_m)$ to guide the search, where $w : S \mapsto \mathcal{W}$ measures the novelty of a state, $\mathcal{W} \in \mathbb{N}$ is the set of novelty categories and $H = \{h_1, \ldots, h_m\}$ is a set of heuristic functions. BFWS algorithm sorts the nodes in order of importance using the first function in $f(n)$, recursively breaking ties using the next function provided in $f(n)$. The *approximation* of BFWS (Singh et al. 2021) uses the same evaluation function to guide the search with only two differences (1) $f(n) = (\hat{w}, h_1, \ldots, h_m)$, where $\hat{w} : S \mapsto \mathcal{W}$ is a function measuring the *approximate* novelty, (2) it uses an adaptive policy, derived from the analytical solution to an infinite-horizon Markov Decision Problem (MDP), that decides whether to forgo the expansion of nodes in the open list (Singh et al. 2021). These improvements result in a state-of-the-art BFWS planner over IPC *satisficing* benchmarks by simply pairing novelty measure with goal-counting heuristic $\#g$, i.e. $f(n) = (\hat{w}, \#g)$.

Sequential *polynomial approximate* BFWS(f_5)

In this planner, we make sequential calls to the *polynomial approximate* BFWS(f_5) (Singh et al. 2021) with novelty based pruning until we run out of time. Each *polynomial approximate* BFWS(f_5) is denoted by Singh et al. (2021) as p-BFWS$(f_5)_{\bar{\omega}}$AC, where the set of novelty categories considered in the computation of novelty is $\mathcal{W} = [1, \bar{\omega} + 1]$ and nodes with $\hat{w}(n) > \bar{\omega}$ are pruned. We denote the sequential configuration as 'pI-BFWS(f_5)AC', where p stands for novelty based *pruning*, I for iterative, A for novelty *approximation* and C for adaptive *control* of the open-list.

We start by calling 'p-BFWS$(f_5)_{\bar{\omega}}$AC' with $\bar{\omega} = 1$, i.e. nodes with $\hat{w}(n) > 1$ are pruned. At each subsequent call,

	Folding	Labyrinth	Quantum L.	Recharging R.	Ricochet R.	Rubik's C.	Slitherlink
ApxNovelty (*agl*)	4	11(0)	20	10 (6)	14	4	3
ApxNovelty (*sat*)	5	15(0)	20	11(8)	18	5	4

Table 1: Coverage of Sequential polynomial Approximate BFWS(f_5) (ApxNovelty) in Agile and Satisficing tracks. The numbers in brackets represent the coverage in the IPC 2023 for domains that encountered preprocessing error[1].

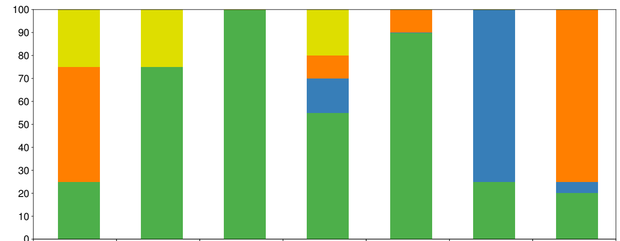
we increase the novelty bound $\bar{\omega}$ by 1. At small values of $\bar{\omega}$ the planner taps into the low polynomial search space of BFWS(f_5) with a small probability of error in the accuracy of *novelty* computation. As the value of $\bar{\omega}$ grows large it becomes harder to compute novelty ω *exactly* for the same large values. Indeed, the original BFWS(f_5) would exceed the space and time limits for $\omega > 2$ on many IPC benchmark domains. p-BFWS$(f_5)_{\bar{\omega}}$AC allows us to tap into that space by trading off the accuracy of novelty computation for time and space guarantees. We have entered this planner into *agile* and *satisficing* track, with one difference in the *satisficing* submission - once 'pI-BFWS(f_5)AC' finds a solution we call the implementation of weighted A* used in LAMA (Richter and Westphal 2010) to improve the plan quality until timeout.

Empirical Analysis [1]

The International Planning Competition of 2023 included many entirely new domains, not seen in the previous IPCs, including Labyrinth - a game where the agent must escape from a maze, Quantum Circuit - which requires the solver to map logical quantum circuits to physical qubits, Recharging Robots - a coordination problem that requires observation robots to schedule their recharging times such that the security levels are maintained, and lastly, the classic Rubik's Cube. The varying characteristics of the domains, *hard-to-ground*, *impractical novelty value of states in feasible plans*, and the complex structures of dependencies between the fluents of states in feasible plans presented a challenge to our planners. In this section, we summarize the results and attempt to justify the observed performance in each domain.

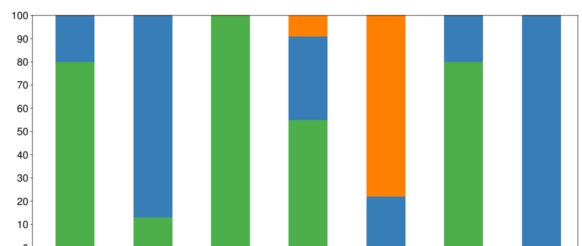
Table 1, which shows the coverage of Sequential polynomial approximate BFWS on the IPC 2023 instances, reveals that increasing the time limit from 300 seconds in *Agile* track to 30 minutes in *Satisficing* only slightly improves the coverage, in 5 of the 7 domains. To further examine the *Satisficing* results, we use two bar plots to illustrate the characteristics of the planning runs. Figure 1 presents the performance profile of the planner in terms of the percentage of the problems solved in each domain and a breakdown of the reason for failure in unsolved instances - load memouts(lm%), load timeouts(lt%), search memouts(sm%), and search timeouts(st%). The figure helps us identify which component of the planner's algorithm stack is challenged by a specific domain. Figure 2 shows the distribution of minimum novelty bound of the polynomial planner at which it finds a solution to the instance. We now explain the performance of the planner on individual domains.

[1]Our integration with FAST-DOWNWARD Grounder failed in Labyrinth and Recharging-robots instances with *single-goal atoms*, resulting in the planner terminating unexpectedly at the step when the search engine's data structures are initialized. The error prevented us from accurately analyzing and justifying the performance of Approximate Novelty Search using the IPC 2023 results. Hence, we redid the experiments on the two domains. We executed the experiments on a server using Intel Xeon Processors (2 GHz).



	Folding	Labyrinth	Quantum	Recharge.	Ricochet.	Rubik's.	Slitherlink
lm%	25	25	0	20	0	0	0
lt%	0	0	0	0	0	0	0
sm%	50	0	0	10	10	0	75
st%	0	0	0	15	0	75	5
s%	25	75	100	55	90	25	20

Figure 1: Plot showing the *performance profile* of the Sequential polynomial Approximate BFWS. s% represents the percentage of solved instances, and the exit codes of unsolved instances are captured as - load memouts(lm%), load timeouts(lt%), search memouts(sm%), search timeouts (st%), where "load" refer to the preprocessing phase of *parsing and grounding*.



	Folding	Labyrinth	Quantum	Recharge.	Ricochet.	Rubik's.	Slitherlink
($\bar{\omega} \geq 3$)%	0	0	0	9	78	0	0
($\bar{\omega} = 2$)%	20	87	0	36	22	20	100
($\bar{\omega} = 1$)%	80	13	100	55	0	80	0

Figure 2: Plot showing the distribution of *minimum novelty bound* in the *polynomial* Approximate BFWS, *necessary* to find a feasible plan in the *solved* instances.

Quantum Layout domain received the Outstanding Domain Submission Award, out of all the submitted domains in the IPC 2023. It is an exciting application of planning on a problem of practical significance. Our planner performs best in this domain, among all the participants of IPC 2023 in the *Agile track*. A detailed analysis of the planner's performance reveals that this is not a coincidence. Our polynomial planner solved 95% instances with a novelty bound of 1. The sole outlier only required one more iteration of the polynomial planner with a bound of 2. Such a consistent finding across all 20 instances suggests that the structure of the instances - reachability relation between fluents in plans - exhibits characteristics that align with the concept of problem width (Lipovetzky and Geffner 2012) in width-based planning algorithms. Since this is a problem of practical interest, we believe that further study to explore the possibility of a low upper bound on the value of novelty measure is warranted.

Ricochet-robots is another domain where the polynomial novelty planner performs well. However, in contrast to Quantum Layout, this problem domain requires expanding nodes of novelty greater than 2. As noted in the paper on Approximate Novelty Search (Singh et al. 2021), the computation of novelty > 2 in the IPC instances is generally impractical, except for tiny instances. However, the fact that we solve 95% of the instances in this domain demonstrates the usefulness of approximation methods that trade-off accuracy for computational guarantees. Here, the *novelty approximation* enables the planner to compute novelty measure *approximately* but in *linear* time and consuming a fixed amount of memory, when it is practically infeasible to do so precisely. In the vast majority of instances of Ricochet-robots, the lower bound on the highest novelty of a state in any plan is 3. This finding is noteworthy as Ricochet-robots is the only domain from the IPCs where the majority of instances require novelty computation that is impractical to do *exactly*, and at the same time, the lower bound on the highest novelty of states in any plan is reachable with Sequential polynomial Approximate BFWS.

Labyrinth is a big instance, also considered to be *hard-to-ground*, in which half of the instances have more than half a million grounded actions. Hence, it is not surprising that many instances failed in the grounding phase. The polynomial novelty planner solved the remaining instance, which could be grounded, with the novelty bound of 2.

In the instances of **recharging robots**, our planner solved 50%[1] of the instances. All except one instance were solved within the novelty bound of 2, and the outlier with a novelty bound of 3. Most instances that could not be solved ran out of memory, likely because of the large number of ground actions, which sometimes exceeded a million.

In **Folding, Rubik's Cube, and Slitherlink**, many instances ran out of computational resources of time and memory while searching for a solution at or below the novelty bound of 3. This observation points to the possibility that the region of state space that is reachable at low novelty bounds of 1, 2, and 3 is significantly large and stresses the open and closed lists - data structures that store a representation of the explicit search tree and the explored state space for us.

Overall, the results show that our planners performed excellently in the IPC 2023 instances, particularly in the Agile Track. Despite the bug[1] that eliminated the coverage on Labyrinth and significantly reduced the coverage on Recharging Robots, the planner still managed to rank seventh out of twenty-three planners. Looking at the new results, we believe that the planner would have ranked among the top two in the Agile Track if not for the unfortunate bug that affected the FD Grounding and LAPKT integration.

Conclusion

The empirical evidence of the planner's performance from the International Planning Competition 2023 is very informative, it provides a holistic picture of the strengths and limitations of Approximate Novelty Search. It provides us the first empirical evidence that the novelty approximation is of real value in domains where states in the plans have novelty values that are impractical to compute exactly but small enough for Sequential polynomial BFWS to manage. Moreover, the algorithm's excellent performance in the *Agile* track demonstrates that probabilistically complete search algorithms are a promising candidate for planning in environments of hard limits on time and memory.

References

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). *Ninth International Planning Competition (IPC 2018), Deterministic Part* .

Katz, M.; Lipovetzky, N.; Moshkovich, D.; and Tuisov, A. 2018. Merwin planner: Mercury enchanced with novelty heuristic. *IPC 2018 planner abstracts* 53–56.

Lipovetzky, N.; and Geffner, H. 2012. Width and Serialization of Classical Planning Problems. In *Proc. of European Conference in Artificial Intelligence (ECAI)*.

Lipovetzky, N.; and Geffner, H. 2014. Width-based algorithms for classical planning: New results. In *Proc. of European Conference in Artificial Intelligence (ECAI)*, 1059–1060. IOS Press.

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)*.

Lipovetzky, N.; Ramirez, M.; Muise, C.; and Geffner, H. 2014. Width and inference based planners: Siw, bfs (f), and probe. *Proc. of the Int'l Planning Competition (IPC)* 43.

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.