Best-First Width Search in the IPC 2018: Complete, Simulated, and Polynomial Variants

Guillem Francès
University of Basel
Basel, Switzerland
first.last@unibas.ch

Hector Geffner
ICREA & U. Pompeu Fabra
Barcelona, Spain
first.last@upf.edu

Nir Lipovetzky
University of Melbourne
Melbourne, Australia
first.last@unimelb.edu.au

Miquel Ramírez
University of Melbourne
Melbourne, Australia
miguel.ramirez@unimelb.edu.au

Abstract

Width-based search algorithms have recently emerged as a simple yet effective approach to planning. Best-First Width Search (BFWS) is one of the most successful satisficing width-based algorithms, as it strikes a good balance between an effective exploration based on a measure of *state novelty* and the exploitation provided by traditional goal-directed heuristics. Several conceptually interesting BFWS variants have recently been shown to offer state-of-the-art performance, including a *polynomial-time* BFWS planner which is incomplete but fast and effective, and a black-box BFWS planner that can plan efficiently with *simulators*, i.e. when the transition function of the problem is represented as a black-box function. In this paper, we describe six BFWS planners involving these variations that we have entered into the 2018 International Planning Competition.

Introduction

Planning as heuristic search is one of the most successful computational approaches to classical planning developed so far (Bonet and Geffner 2001; Hoffmann and Nebel 2001), dominating several of the past editions of the International Planning Competition (IPC). The essential component of this approach is the *automatic derivation* of an heuristic function that informs the search from the declarative representation of the problem in some modeling language such as STRIPS or PDDL (Fikes and Nilsson 1971; McDermott 2000). This is usually coupled with a suitable search strategy and a number of search improvements such as helpful actions, delayed evaluation and multiple search queues (Hoffmann and Nebel 2001; Helmert 2006).

A recent and significant departure from this approach are *width-based search algorithms*. Still framed within the *planning as search* paradigm, width-based algorithms however do away with the reliance on heuristics and *means-ends* analysis (Newell and Simon 1963), and use instead a powerful exploration mechanism based on a structural, goal-agnostic notion of *state novelty*, which roughly assigns value to states based on how novel they are with respect to the states already visited by the search strategy being employed (Lipovetzky and Geffner 2012). The exploration mechanism offered by the width-based approach can be combined with traditional heuristics in greedy best-first-like search strategies to produce state-of-the-art satisficing planning strategies collectively called Best-First Width Search (BFWS) (Lipovetzky and Geffner 2017a; Katz *et al.* 2017), but it also has other interesting properties that go beyond performance. First, state novelty measures seem to be a particularly effective pruning mechanism. Lipovetzky and Geffner (2017b) have developed *incomplete but polynomial* BFWS variations with a simple modification that consists on roughly pruning from the search those nodes which are not novel enough. The resulting algorithm solves more instances from previous competitions than IPC-winning, exponential-time planners such as LAMA or FF (Richter and Westphal 2010; Hoffmann and Nebel 2001).

Second, width-based methods constitute a surprising departure from previous planning research, as they do not require a declarative definition of the action model, an essential component of virtually all previous approaches, from the first means-ends and partial-order planners (Newell and Simon 1963; Tate 1977; Nilsson 1980) to the latest SAT, OBDD, and heuristic search planners (Kautz and Selman 1996; Edelkamp and Kissmann 2009; Richter and Westphal 2010; Rintanen 2012). Francès *et al.* (2017) show how width-based methods are an effective means of dealing with the standard IPC benchmarks even when *no information on the action structure is available* to be used in the computation of e.g. heuristics or SAT models, that is, when the transition function of the problem is given as a black box. This is relevant, as there is a wide set of problems which fit the classical planning model, but whose dynamics are not easily represented in declarative languages, cf. the Atari Learning Environment (Bellemare *et al.* 2013), the games of the General Video Game competition (Perez-Liebana *et al.* 2016), Angry Birds (Renz 2015) and Minecraft (Johnson *et al.* 2016), all of which expose action models through procedural, black-box interfaces that preclude the use of most classical planners. At the same time, having effective planning algorithms that do not rely on a declarative representation of the action model greatly reduces the challenge of modeling, as arbitrary, high-level language constructs such as axioms or semantic attachments (Thiébaux *et al.* 2005; Dornhege *et al.* 2009) can be seamlessly dealt with.

Width-based methods have also recently been extended beyond classical planning to tackle finite horizon MDPs (Geffner and Geffner 2015), partially observable problems (MacNally *et al.* 2018) and problems with hybrid discrete and continuous dynamics (Ramirez *et al.* 2018). We here focus on the width-based classical planners that we have submitted to this year's International Planning Competition.

The remainder of this paper is organized as follows. We first present the essential ideas of width-based search, then briefly describe the Best-First Width Search (BFWS) framework, and highlight two interesting possibilities of BFWS-derived planners which are used in some of our submitted planners: polynomial runtime guarantees which make BFWS incomplete but still quite effective, and the ability of planning effectively on black-box representations of the transition functions. We conclude by briefly reviewing the actual planners entered into the competition. Many details have been omitted for the sake of brevity, but we provided pointers to the relevant literature where necessary.

Width-Based Search

Width-based search algorithms are forward state-space search algorithms that rely on the key notion of *novelty* of a state (Lipovetzky and Geffner 2012). Assuming that a state is a set of propositional atoms, as standard in STRIPS-based classical planning, the novelty $w(s)$ of a state s is the size of the smallest set of atoms Q such that s is *the first state encountered in the search* where $Q \subseteq s$. Thus, if s is the first state on the search that contains a certain atom p, then $w(s) = 1$. If no such atom exists, but s is the first state on the search that contains a certain pair of atoms $\{p, q\}$, then $w(s) = 2$, and so on. An important property of the novelty of a state is that it is a search-dependent but goal-independent measure whose computation requires only knowledge about the structure of the state.

The simplest width-based algorithm is the parametric IW(k), a standard breadth-first search where any newly-generated state s with novelty $w(s) > k$ is pruned (Lipovetzky and Geffner 2012). IW(k) converges to breadth-first search as the value of k approaches the number n of atoms in the problem, but its time and space complexity are exponential only in k, hence polynomial if we consider a fixed value of k. Interestingly, IW(k) has been shown to solve *any instance* of many of the standard benchmark domains with $k = 2$, i.e. in quadratic time, provided that the goal is a single atom (Lipovetzky and Geffner 2012; Lipovetzky 2014). This is because such domains have a small and bounded *width* ω that does not depend on the size of the instance and such that IW(k) with $k = \omega$ can (optimally) solve any of their instances.

When goals are however not restricted to single atoms but can be arbitrary conjunctions, strategies more sophisticated than IW(k) are necessary. Different width-based algorithms have been proposed to address that challenge, such as *Serialized IW* (SIW) (Lipovetzky and Geffner 2012), SIW^+ or DFS^+ (Lipovetzky and Geffner 2014). The most successful among these approaches, which we describe next, is the generic search schema known as Best-First Width Search.

Best-First Width Search

Lipovetzky and Geffner (2017a) have recently shown that state-of-the-art performance over the standard classical planning benchmarks can be achieved when the exploration afforded by structural measures of width is combined with the exploitation offered by traditional heuristic search methods. BFWS is a standard best-first search that uses an extended definition of *novelty* of a search node *given certain partitioning functions* as the main criterion to prioritize nodes in the open list. The novelty $w(s)$ of a state s *given functions* $h_1, \dots, h_n$ is defined as the size of the smallest set of atoms Q such that s is the first state encountered in the search where all atoms in Q are true at the same time, *considering only those previous states* s' *with equal* h_i*-values*, i.e., such that $h_i(s) = h_i(s')$ for $i = 1, \dots, n$. This novelty measure is also written as $w_{\langle h_1, \dots, h_n \rangle}(s)$.

The best-performing BFWS planner described in (Lipovetzky and Geffner 2017a) is BFWS(f_5), which uses $w = w_{\langle \#g, \#r \rangle}$, where $\#g(s)$ counts how many of the atomic goals of the problem are not true in s, and $\#r(s)$ is a path-dependent approximation of progress towards achieving a certain set $R(s)$ of atoms which are considered to be relevant to reach the problem goal from state s. A complete description of the algorithm can be found in (Lipovetzky and Geffner 2017a); for the sake of brevity, we here simply note that different alternatives in defining $R(s)$ are possible; the one that works best in (Lipovetzky and Geffner 2017a) is computed from a delete-free relaxed plan computed from s (Hoffmann and Nebel 2001).

Polynomial BFWS

Best-First Width Search provides an alternative to the IW(k) and Serialized IW(k) algorithms (Lipovetzky and Geffner 2012) which is *complete*, but this necessarily implies that the polynomial-time nature of IW(k) is lost. Lipovetzky and Geffner (2017b) present additional variations of BFWS which have guaranteed polynomial runtime, and in spite of being incomplete, can solve a surprising amount of the classical planning benchmarks from previous IPCs. The first such variant is k-BFWS, which is equal to BFWS but prunes from the search those generated states s with novelty $w(s) > k$. The second variant, k-M-BFWS, relaxes that strict pruning criterion by allowing for the expansion of at most M states with novelty higher than k, provided that they are direct descendants of some state s with novelty $w(s) < k$ (Lipovetzky and Geffner 2017b).

Black-Box BFWS

One of the properties of width-based algorithms is that the computation of novelty they rely on only requires knowledge about the structure of the state, not about the action model of the problem. This has allowed the successful use of this approach in simulated environments such as the Atari Learning Environment (Bellemare *et al.* 2013), where a declarative definition of the action model is not available or would be much harder to obtain than a procedural, black-box implementation of the transition function of the problem (Lipovetzky *et al.* 2015; Shleyfman *et al.* 2016).

On this same line, Francès *et al.* (2017) present BFWS(R), a generalization of the BFWS(f_5) algorithm described above which uses alternative strategies for computing the sets $R(s)$ *that do not require a declarative representation of the action model*. The basic idea in most of these strategies is to run a polynomial preprocessing phase where the IW(1) (and, if necessary, IW(2)) algorithm is run from the initial state of the problem to conduct an exploration of the novelty-1 (and, eventually, novelty-2) polynomial subspace of the state space that allows us to identify which of the problem atoms lie on some path that reaches at least some of the atoms in the goal conjunction. The key assumption behind this strategy is that the problem goal is expressed as a conjunction of atoms, and that most goal atoms can be *individually* reached by the polynomial IW(1) or IW(2) algorithms,[1] assumptions which many of the benchmarks from past IPCs share.

Competition Planners

We here briefly describe the characteristics of the 6 different width-based planners submitted to the competition. Table 1 summarizes the properties of each of the submitted planners in terms of completeness, complexity bounds, and their ability to deal with simulators, i.e. black-box representations of the transition function. Our two simulation-based planners, FS-blind and FS-sim, use the PDDL action model of the problem only at preprocessing, to (1) compile an efficient black-box representation of the transition function based on Fast Downward's *successor generator* (Helmert 2006), and (2) perform an ASP-based reachability analysis through the Clingo ASP solver (Gebser *et al.* 2012). The fact that these planners completely ignore the action model after this preprocessing can be seen as an unnecessary handicap in the context of the competition, but we are interested in observing the actual performance of this strategy, which has interesting applications beyond PDDL-based planning. All planners are implemented mostly in `C++`, with some parsing and preprocessing implemented in `Python`, and are built on top of the `LAPKT` planning toolkit (Ramirez *et al.* 2015).

BFWS-*preference* This is the BFWS(f_5) planner described above (Lipovetzky and Geffner 2017a), with one difference for the *satisficing* track submission: once BFWS(f_5) finds a solution, the plan cost is given as an upper bound to the weighted A* (WA*) implementation used in LAMA (Richter and Westphal 2010), which then runs to optimize solution quality until the timeout is reached. The problem given to WA* is preprocessed by h^2 to reduce the number of actions and minimize the search effort (Alcázar and Torralba 2015). This planner has been submitted to the *agile* and *satisficing* tracks.

BFWS-*polynomial* This is the polynomial k-BFWS(f_5) (Lipovetzky and Geffner 2017b), which runs BFWS(f_5) but prunes those nodes whose novelty is higher than k. The planner runs 1-BFWS first; if no solution is found, then a sequence of 2-M-BFWS calls with $M = 1, 2, 4, 8, 16, 32$ follows, where M is a parameter that stands for how many children n' of any node n with novelty $w(n) \leq 2$ have themselves novelty $w(n') > 2$ but are *not pruned*. To keep the submission polynomial, no optimization step has been used on the *satisficing* track. This planner has been submitted to the *agile* and *satisficing* tracks.

Dual-BFWS Dual-BFWS (Lipovetzky and Geffner 2017a) uses the polynomial and incomplete 1-BFWS search, pruning all nodes whose novelty is bigger than 1. If this incomplete search fails, a complete BFWS(f_6) search is run, where $f_6 = \langle w_{\langle h_L \rangle}, help, h_L, w_{\langle h_{\mathrm{FF}} \rangle}, h_{\mathrm{FF}} \rangle$ combines novelty measures with the landmark-based h_L heuristic (Richter and Westphal 2010), helpful actions and h_{FF} (Hoffmann and Nebel 2001). This type of dual architecture is present in early successful planners such as FF. This planner has been submitted to the *agile*, *satisficing* and *cost-bounded* tracks. The *satisficing* track submission includes the same WA*-based optimization as described for the BFWS-*preference* planner, whereas the *cost-bounded track* submission just uses the bound to prune solutions while searching.

DFS$^+$ This is the extension of SIW$^+$ described in (Lipovetzky and Geffner 2014), but instead of increasing the bound of IW(k) until a new goal is reached or the problem is solved, when the bound is 2, we backtrack to the last IW search and continue searching for other states that achieve one more goal. DFS$^+$ can be approximated as a BFWS(f_5) where the evaluation function is reversed as $f_5 = \langle \#g, w_{\langle \#r, \#g \rangle} \rangle$, i.e. preferring first states that achieve more goals, and then breaking ties by novelty extended with the goal and relax plan counters. DFS$^+$ is polynomial, but the number of nodes it expands depends on the number of states that decrease the count $\#g$ of unachieved goal atoms within each IW$^+$(2) call, which is hard to estimate. This planner has been submitted to the *agile* and *satisficing* tracks, with no optimization step for the latter.

FS-blind FS-blind is the BFWS(R_0) simulation-based planner as described in (Francès *et al.* 2017). The planner runs a Best-First Width Search where the set of relevant atoms $R(s)$ is always taken to be the empty set, which effectively means that the only information about the goal that the planner exploits is a simple goal-count heuristic that evaluates how many atoms in the goal conjunction are satisfied in each state. This planner has been submitted to the *agile* and *satisficing* tracks.

FS-sim FS-sim is the BFWS(R^*_G) simulation-based planner as described in (Francès *et al.* 2017), which is like FS-blind above but exploits additional information about the goal, inferred in an extra preprocessing step by running the IW(1) and IW(2) algorithms from the initial state of the problem (as described in the *Simulation-Based BFWS* section above). This planner has been submitted to the *agile* and *satisficing* tracks.

[1] Note that finding plans that reach each of the goals individually is different than finding plans that reach all goals *jointly*.

Planner	Complete	Polynomial	Black-box
BFWS-preference	x		
BFWS-polynomial		x	
Dual-BFWS	x		
DFS$^+$		x	
FS-blind	x		x
FS-sim	x		x

Table 1: Properties of width-based planners submitted to IPC 2018.

Language Expressivity

All of the planners submitted to the competition support universal quantification and conditional and universally-quantified effects, but do not have full support for axioms. The two BFWS(R)-based planners (FS-blind, FS-sim) are implemented in the `FS` planner (Francès and Geffner 2015; 2016), which deals with problems specified in the Functional STRIPS language (Geffner 2000), a superset of STRIPS with support for function symbols. The planner additionally supports a number of interesting extensions which inspired the development of the BFWS(R) algorithms, such as black-box specifications of the transition function, or of the procedural denotation of certain fixed predicate and function symbols, also referred to as semantic attachments (Dornhege *et al.* 2009)

The rest of the submitted planners (BFWS-*preference*, BFWS-*polynomial*, Dual-BFWS and DFS$^+$) support conditional and universally-quantified effects, as said, but their use of the Fast Downward parser, which occasionally transforms universal quantifiers into axioms, make these planners fail in these rare cases, since support for axioms is not yet implemented.

Summary

We have presented the six satisficing classical planners submitted to the 2018 International Planning Competition. These six planners explore different conceptually interesting variants of best-first width search: some are complete, some are incomplete but polynomial, and some are black-box procedures that do not require any declarative representation of actions in terms of preconditions and effects.

Acknowledgments

M. Ramírez and N. Lipovetzky have been partially funded by DST.

References

V. Alcázar and A. Torralba. A reminder about the importance of computing and exploiting invariants in planning. In *ICAPS*, pages 2–6, 2015.

M. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents. *JAIR*, 47, 2013.

B. Bonet and H. Geffner. Planning as heuristic search. *Artificial Intelligence*, 129(1–2):5–33, 2001.

C. Dornhege, P. Eyerich, T. Keller, S. Trüg, M. Brenner, and B. Nebel. Semantic attachments for domain-independent planning systems. In *Proc. ICAPS*, 2009.

S. Edelkamp and P. Kissmann. Optimal symbolic planning with action costs and preferences. In *Proc. IJCAI*, 2009.

R. Fikes and N. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 1:27–120, 1971.

G. Francès and H. Geffner. Modeling and computation in planning: Better heuristics for more expressive languages. In *Proc. ICAPS*, 2015.

G. Francès and H. Geffner. E-STRIPS: Existential quantification in planning and constraint satisfaction. In *Proc. 25th Int. Joint Conf. on Artificial Intelligence*, page 3082, 2016.

G. Francès, M. Ramırez, N. Lipovetzky, and H. Geffner. Purely declarative action representations are overrated: Classical planning with simulators. In *Proc. IJCAI*, 2017.

Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. Answer set solving in practice. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(3):1–238, 2012.

T. Geffner and H. Geffner. Width-based planning for general video-game playing. In *Proc. AIIDE*, 2015.

H. Geffner. Functional STRIPS. In J. Minker, editor, *Logic-Based Artificial Intelligence*, pages 187–205. Kluwer, 2000.

M. Helmert. The Fast Downward planning system. *JAIR*, 26:191–246, 2006.

J. Hoffmann and B. Nebel. The FF planning system: Fast plan generation through heuristic search. *JAIR*, 14:253–302, 2001.

M. Johnson, K. Hofmann, T. Hutton, and D. Bignell. The Malmo platform for AI experimentation. In *Proc. IJCAI*, 2016.

M. Katz, N. Lipovetzky, D. Moshkovich, and A. Tuisov. Adapting novelty to classical planning as heuristic search. In *Proc. ICAPS*, 2017.

H. Kautz and B. Selman. Pushing the envelope: Planning, propositional logic, and stochastic search. In *Proc. AAAI*, 1996.

N. Lipovetzky and H. Geffner. Width and serialization of classical planning problems. In *Proc. ECAI*, 2012.

N. Lipovetzky and H. Geffner. Width-based algorithms for classical planning: New results. In *Proc. ECAI*, pages 88–90, 2014.

N. Lipovetzky and H. Geffner. Best-first width search: Exploration and exploitation in classical planning. In *Proc. AAAI*, 2017.

N. Lipovetzky and H. Geffner. A polynomial planning algorithm that beats lama and ff. In *Proc. ICAPS*, 2017.

N. Lipovetzky, M. Ramirez, and H. Geffner. Classical planning with simulators: Results on the atari video games. In *Proc. the Joint International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

N. Lipovetzky. *Structure and Inference in Classical Planning*. AI Access, 2014.

A. M. MacNally, N. Lipovetzky, M. Ramirez, and A. R. Pearce. Action selection for transparent planning. In *Proc. AAMAS*, 2018.

D. McDermott. The 1998 AI Planning Systems Competition. *AI Magazine*, 21(2):35–56, 2000.

A. Newell and H. Simon. GPS: a program that simulates human thought. In E. Feigenbaum and J. Feldman, editors, *Computers and Thought*, pages 279–293. McGraw Hill, 1963.

N. Nilsson. *Principles of Artificial Intelligence*. Tioga Press, 1980.

D. Perez-Liebana, S. Samothrakis, J. Togelius, T. Schaul, and S.M. Lucas. General video game AI: Competition, challenges and opportunities. In *Proc. AAAI*, 2016.

M. Ramirez, N. Lipovetzky, and C. Muise. Lightweight Automated Planning ToolKiT. `http://lapkt.org/`, 2015.

M. Ramirez, M Papasimeon, N. Lipovetzky, L. Benke, T. Miller, A. R. Pearce, E. Scala, and M. Zamani. Integrated hybrid planning and programmed control for real–time uav maneuvering. In *Proc. AAMAS*, 2018.

J. Renz. AIBIRDS: The Angry Birds artificial intelligence competition. In *Proc. AAAI*, 2015.

S. Richter and M. Westphal. The LAMA planner: Guiding cost-based anytime planning with landmarks. *JAIR*, 39:122–177, 2010.

J. Rintanen. Planning as satisfiability: Heuristics. *Artificial Intelligence*, 193:45–86, 2012.

A. Shleyfman, A. Tuisov, and C. Domshlak. Blind search for atari-like online planning revisited. In *Proc. IJCAI*, 2016.

A. Tate. Generating project networks. In *Proc. IJCAI*, 1977.

S. Thiébaux, J. Hoffmann, and B. Nebel. In defense of pddl axioms. *Artif. Intell.*, 168(1-2):38–69, 2005.