Hierarchical Planning in the IPC

Unknown Authors

Abstract

Over the last year, the amount of research in hierarchical planning has increased, leading to significant improvements in the performance of planners. However, the research is diverging and planners are somewhat hard to compare against each other. This is mostly caused by the fact that there is no standard set of benchmark domains, nor even a common description language for hierarchical planning problems. As a consequence, the available planners support a widely varying set of features and (almost) none of them can solve (or even parse) any problem developed for another planner. With this paper, we propose to create a new track for the IPC in which hierarchical planners will compete. This competition will result in a standardised description language, broader support for core features of that language among planners, a set of benchmark problems, a means to fairly and objectively compare HTN planners, and for new challenges for planners.

Introduction

When the International Planning Competition (IPC) started out in 1998 it aimed to include both classical and hierarchical planners as competitors (McDermott 2000). To provide a fair competition and foster further research in planning, a unified description language for planning problems, the Planning Domain Definition Language (PDDL), was created. The main objective of PDDL was to separate the description of the physics of a domain from additional advice. Planners should operate on the physics of the domains in the IPC. This way the competition shows how good the underlying domain-independent techniques are for solving planning problems based on the description of a domain's physics. Notably, PDDL also included features for expressing hierarchical planning problems. Hierarchical planning was not included in the first IPC, as there were no competitors.

The second and third IPC subsequently included a separate track for "Hand-Tailored Systems", where the planner was allowed to have additional advice for each individual domain. This was within the spirit of hierarchical planning at the time (see e.g. SHOP's participation (Nau et al. 1999)). Hierarchical structures in a planning problem were – at the time – almost exclusively seen as additional advice given to the planner in order to speed up planning. The hierarchy was viewed as specifying pre-defined recipes for a given objective. After the third IPC, no further attempt was made to include hierarchical planning into the IPC.

Since the third IPC in 2002, the research in hierarchical planning has progressed significantly. Especially the view what hierarchical planning is, has changed sharply. A hierarchy over a planning problem is not (only) a means to provide advice to the planner, but a general means to specify parts of the domain. Just like non-hierarchical structures, it may be used to model advice (as done before), but also to describe additional physics of the domain. Research in hierarchical planning has, over the last decade, mostly focussed on a single formalism: Hierarchical Task Network (HTN) Planning (Erol, Hendler, and Nau 1996)[1]. The physics expressible in HTN planning cannot be equivalently expressed in classical planning (Höller et al. 2014). In fact, HTN planning allows for expressing undecidable problems, like Post's Correspondence Problem or Context-free Grammar Intersection (Erol, Hendler, and Nau 1996). Based on the researcher's focus on a single, theoretically understood formalism, we think the time is right for an IPC track in which HTN planners compete. Creating such a competition was one of the motivations behind organising the Workshops on Hierarchical Planning at ICAPS 2018 and 2019. A discussion on the details of a (potential) IPC track for HTN planning is planned for this year's workshop.

The HTN planning community lacks a strong unifying force like the IPC is for classical planning. Due to the lack of a competition in HTN planning, there is no precisely agreed-upon semantics for modelled planning problems[2], no unified input language readable by all planners, and no standardised benchmark to compare planners. A competition should resolve these problems and provide a solid basis for future work, as was the case for classical planning. We therefore propose to add a new track to the IPC in which HTN planners will compete. In this paper, we first give a brief overview of HTN planning, report on results of an online questionnaire on an HTN IPC track, and then present the details of our prosed HTN IPC track.

Copyright © 2019, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

[1]Note that there is also research within hierarchical planning that does not qualify as HTN planning, such as work on HGN and GTN planning (Shivashankar et al. 2013; Alford et al. 2016b)

[2]Although there is a theoretical formalism almost all planners add their individual bells and whistles making them incomparable.

HTN Planning – State of the Art

We start by giving a quick overview over HTN planning, theoretical and practical results related to it.

Theory

While classical planning has only one type of tasks – actions – HTN planning distinguishes two types: primitive tasks (also called actions) and abstract tasks. Only actions hold preconditions and effects, which are defined in the usual (STRIPS or ADL) way. In contrast to actions, abstract tasks cannot be executed directly. Instead, each abstract task A has a set of associated decomposition methods (A, tn) that allow for achieving A by performing the tasks (that may, again, be primitive or abstract) contained in tn. Here tn is a task network, which gives HTN planning its name. A task network is a partially ordered multi-set of tasks.

The objective in HTN planning is given by an initial state and an initial abstract task (replacing the state-based goal definition in classical planning[3]). A solution is any sequence of actions that is executable in the initial state, provided it can be obtained from the initial abstract task by repeatedly applying decomposition methods. In this structure, HTN planning is very similar to formal languages, where terminals correspond to actions and non-terminals to abstract tasks (Höller et al. 2014; Barták and Maillard 2017).

A simplistic formal description of HTN planning was proposed by Geier and Bercher (2011). An equivalent formulation in terms of combinatorial grammars was given by Barták, Maillard, and Cardoso (2018). The simplistic formalism by Geier and Bercher is theoretically well understood, witnesses by the multitude of papers based on it.

The plan existence problem in HTN planning in its general form is undecidable (Erol, Hendler, and Nau 1996). There are, however, several restrictions to the formalism that make it decidable. The most notable one is totally-ordered HTN planning. Here, all task networks in decomposition methods must be sequences of tasks instead of partially-ordered sets. In contrast to general HTN planning, ground, totally-ordered HTN planning is EXPTIME-complete (Erol, Hendler, and Nau 1996).

Planners

In the past, a multitude of HTN planners have been developed. This includes the older systems which were mostly based on uninformed search, like SHOP (Nau et al. 1999), SHOP2 (Nau et al. 2003), or UMCP (Erol, Hendler, and Nau 1994). Recently, the portfolio of available methods to solve HTN planning problems has increased significantly. Many ideas developed for classical planning have been transferred to HTN planning problem and proved successful there. This lead to a wide range of (partially) available HTN planners.

- FAPE (Dvorak et al. 2014), a temporal HTN planner with strong pruning techniques
- PANDA (Bercher et al. 2017), a plan-space planner using heuristic search
- PANDApro (Höller et al. 2018), a progression-based planning system using heuristic search
- GTOHP (Ramoul et al. 2017), a planner based on intelligent grounding and blind search
- HTN2ASP (Dix, Kuter, and Nau 2003), a planner that translates (totally-ordered) HTN planning problems into answer set programming
- HTN2STRIPS (Alford et al. 2016a), a planner translating HTN planning problems into a sequence of classical planning problems
- totSAT (Behnke, Höller, and Biundo 2018) and Tree-Rex (Schreiber et al. 2019), planners that translates (totally-ordered) HTN planning problems into propositional logic
- partSAT (Behnke, Höller, and Biundo 2019), a planner based on a translation into propositional logic

It is noteworthy that four of the planners (GTOHP, totSAT, Tree-Rex, and HTN2ASP) are specifically designed for solving totally-ordered HTN planning problems. Further, HTN2STRIPS is based on an older encoding of totally-ordered HTN planning problems into classical planning (Alford, Kuter, and Nau 2009).

[3]Note that every classical planning problem can be converted into an equivalent HTN planning problem (Erol, Hendler, and Nau 1996).

Online Questionnaire

In order to better understand the needs and interest of the community in organizing a deterministic HTN IPC track in 2020, we conducted a short online questionnaire and sent the link for participation to the planning community.

We obtained 23 responses from 20 different research groups. 80% of the respondents would like to participate to an HTN IPC track in 2020. Those not wishing to participate often justified their refusal with the lack of HTN standards (language, benchmarks, tools). We would argue that starting the competition is the only way to produce such a standard in the first place. As such, organising an HTN IPC competition seems to respond to the needs of the community.

The main motivation for those wishing to participate is to be able to objectively compare the performance of HTN planners. However, this was not the only reason given by respondents. People also wished to increase the expressiveness of the PDDL language by adding HTN language standards.

We also asked what the respondents thought to be absolute necessities for the competition. The four top answers ranked in order of importance are:

1. a standard language definition for HTN planning,
2. an accessible HTN benchmarks repository,
3. an HTN plan validator, and
4. an HTN parser.

In terms of tracks, most people think that optimal and satisfying should be enough for a first IPC competition. Some people also expressed interest in participating in a temporal track. For the first HTN IPC, we propose to restrict the competition to a non-temporal setting, with the option of a temporal track to be added in the future. We think that the competition should focus on the core of HTN planning and compare the planners on the absolute essentials.

In terms of metrics that should be used to compare HTN planners, opinions are divided. 50% of the respondents think

that the IPC score is suitable for an HTN IPC track. 50% think that we need to devise new metrics, which they could propose as free text answers. Respondents proposed to use (1) the number of backtracking operations performed by the planner, (2) the depth the explored search space, (3) the size of the method library used by the planner, and (4) the expressivity and explicability of the solution.

We propose to keep the IPC score as opposed to these proposals. First, we argue that all planners in the competition *must* use the same HTN method library to ensure that the results of the competition are correct, objective, and fair. If every planner would be allowed its own set of methods, we would not measure the performance of the planner, but the ingenuity of the modeller. Furthermore, the set of solutions the problems encodes would differ from planner to planner – which makes little to no sense, as a comparison between the planners is not possible anymore.

Using either the number of backtracking operations or the depth of the explored search space would restrict the competition to search-based planners, as e.g. SAT-based planners can't produce these measures. We think that these two metrics stem from the idea that the planners may use their own HTNs for the domains, as both the number of backtracking operations as well as the depth of the search space are metrics to measure the "difficulty" of an HTN domain. Further, to measure these metrics, we would have to rely on the information reported by the planners about their internal search, which is somewhat strange for an objective competition. Similarly, measuring the explicability of solutions objectively is hard or even impossible at the moment.

In terms of HTN planning language, a majority of people thinks it is better to add hierarchical planning concepts to the PDDL language rather than redefining a new language from scratch. We elaborate on this in the next section.

Concerning benchmarks, most respondents wish to participate to the definition of HTN benchmarks. We asked respondents which types of domains they wished to see as part of the benchmark set (see Fig. 1). Note that these answers have to be considered with care, as most of the mentioned types of domains are also domains used in the classical IPC. HTN planning is *not* a means to solve classical planning problems faster, but a complex combinatorial problem in its own right. There are structural restrictions (e.g. PCP and Grammar Intersection domains) to plans that cannot be expressed by preconditions and effects of actions, while it is possible to formulate them in an HTN domain (Erol, Hendler, and Nau 1996; Geier and Bercher 2011; Höller et al. 2014). Further, the domain's hierarchy can express physics that is not modelled in the primitive actions and is thus an integral part of the problem definition. For example, in a transport domain, the location of each truck can be fully modelled in an HTN problem without introducing state variables pertaining to the location of the truck. Without the hierarchy, the primitive action theory in this model makes no sense at all. As such, the mentioned domain proposals should be viewed as potential topics for modelling domains with highly complex restrictions on plans and not as the suggestion that the domain should express the same mechanics as its classical counterpart.

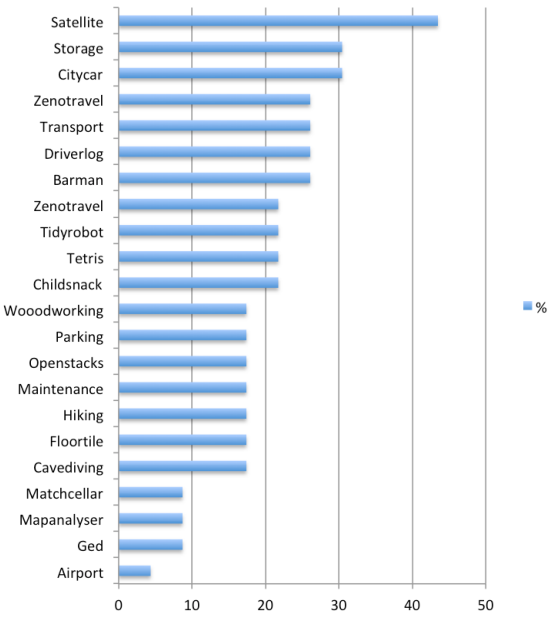

Figure 1: Types of domains proposed by respondents for HTN IPC track

Common Description Language

Organising the first IPC track on HTN planning also involves defining a fixed input language for all participating planners. For classical planning, there is one universally accepted description language – PDDL. The situation for HTN planning is drastically different. Of the planners mentioned in the previous section, PANDA, PANDApro, totSAT, and partSAT accept the same input language, as do GTOHP and Tree-Rex, while all other planners use their individual description languages. Both groups are based upon the same parser and grounder. The two formats are however somewhat incompatible and the format of GTOHP and Tree-Rex does only support totally-ordered HTN planning problems.

Some of the used description languages are based on PDDL, while others, like SHOP's language and ANML (Smith, Frank, and Cushing 2008) are drastically different. We think that a language based on PDDL is the most sensible way to define the HTN description language, as it will allow for close contact between the HTN and classical planning communities. A proposal for an HTN description language has been submitted to this year Workshop on Hierarchical Planning at ICAPS, which will be discussed at the workshop provided the paper will be accepted.

Tracks

As in the IPC for deterministic, classical planning, we propose to run competitions for optimal, satisficing, and agile planning. Further we propose to split the competition between the types of problems handled by the planners. Since there is a large group of existing HTN planners that is restricted to totally-ordered HTN planning problems, we propose to include a separate track for those planners in which

all input problems are totally-ordered. Naturally, all HTN planners that are capable of handling general HTN planning problems can take part in a totally-ordered competition, but should be at a disadvantage. In a second set of sub-tracks, the planners that are able to solve general, i.e. partially-ordered, HTN planning problems will compete. As a result, we propose six tracks: optimal totally-ordered, satisficing totally-ordered, agile totally-ordered, optimal general, satisficing general, and agile general.

Timeline and Organisation

We propose to roughly follow the usual schedule of the IPC.

May – July 2019	Agreeing on a common input language for all planners.
July 2019	Announcement of the track Call for domains Call for expression of interest
October 2019	Registration deadline
November 2019	Demo problems provided
January 2020	Submission of preliminary planner versions
February 2020	Domain submission deadline
April 2020	Final planner submission deadline
May 2020	Paper submission deadline
May 2020	Contest run
June 2020	Presentation of the results at ICAPS 2020

As did the first IPC track on unsolvability (Muise and Lipovetzky 2015), we also expect that there will be many technical issues with the submitted planners. We assume that most of them will be centred around correct support of the input language, for which issues will usually take some time to debug and fix. Further, HTN planners have to provide the decompositions they used as their output as well in order to be able to verify their solutions in time. Without these decomposition, verification is $\mathbb{NP}$-complete (Behnke, Höller, and Biundo 2015) and thus may take a long time. We will use the preliminary submissions of the planners to validate that their outputs are correctly formatted and can be verified against the domain. As such, we propose a first submission of the planners early on, so that we can test them sufficiently before the actual competition.

Scoring and Setting

For scoring the planner, we will adopt the metrics used in the last deterministic, classical IPC. We further propose to use the same technical setting (1 core, 8 GB of RAM, and 30 minutes). All planners will be provided with the same HTN planning domain, i.e. set of primitive actions, abstract tasks, and decomposition methods, the same planning problem, i.e. initial state and initial abstract task, and are not allowed to have any additional domain-dependent information. They have to output both the solution, i.e. a sequence of primitive actions, as well as a witness that this solution is obtainable via decomposition form the initial abstract task.

Domain

As noted before, there is no large set of available benchmarking domains for HTN planners, which would have to be created for the competition. The IPC has so far relied for the domains in the competition on an open call for the community to submit domains. We propose will also call for such community-provided domains (once a input language has been fixed). In addition, we propose to add a new mode of providing benchmark domains that has been recently adopted by the SAT community: Bring Your Own Benchmark (BYOB). The SAT Competition 2017[4] and 2018[5] used BYOB. In it, each competing program – in our case planner – is required to submit a domain with 20 instances. Of these 20 instances, the planner of the submitter must be able to solve at *most* 10. This requirement forces submissions of domains that are not advantageous to the planner of the submitter. This encourages the submission of problems that are of medium difficulty for the individual planner. We argue that they will both provide a good basis for comparison against other planners as well as a good starting point for future scientific investigations based on the difficulties in them.

From a content perspective, we especially encourage submissions of domains that pose problems which cannot be expressed in classical planning, i.e. in PDDL[6]. Such domains exists, like PCP and Grammar Intersection, but it would be interesting to see where HTN planning can further use its high expressive power.

Discussion

With a new IPC track on HTN planning, we think that the research in hierarchical planning will be more focussed and successful in the future. As a result of the competition, we expect that

- the HTN community will agree upon a set of core features supported by every planner and an input language for it, which is important for users of HTN planning, as well as for comparing systems against each other.
- a set of benchmark domains will be available, allowing for better judging and fairer comparisons of planners than is currently possible.
- a core of (relatively) error-free software to be used by many planners will emerge, like Fast Downward (Helmert 2006) for classical planning, allowing for both easy use by planning researchers and users of planning technology.
- hints for future research in HTN planning will be given.

While the first three points are valuable to the community and outsiders wanting to use HTN planning, we want to emphasise the fourth. The results of planning competitions will regularly show the weaknesses of the so-far developed approaches and techniques. These weaknesses are valuable information and point out where planners can and should be improved in the future.

[4] https://baldur.iti.kit.edu/sat-competition-2017/

[5] http://sat2018.forsyte.tuwien.ac.at/

[6] without numbers

References

Alford, R.; Behnke, G.; Höller, D.; Bercher, P.; Biundo, S.; and Aha, D. W. 2016a. Bound to plan: Exploiting classical heuristics via automatic translations of tail-recursive HTN problems. In *Proc. of the 26th Int. Conf. on Autom. Plan. and Sched., (ICAPS 2016)*, 20–28. AAAI Press.

Alford, R.; Shivashankar, V.; Roberts, M.; Frank, J.; and Aha, D. W. 2016b. Hierarchical planning: relating task and goal decomposition with task sharing. In *Proc. of the 25th Int. Joint Conf. on AI (IJCAI 2016)*. AAAI Press.

Alford, R.; Kuter, U.; and Nau, D. 2009. Translating HTNs to PDDL: A small amount of domain knowledge can go a long way. In *Proc. of the 21st Int. Joint Conf. on AI (IJCAI 2009)*, 1629–1634. AAAI Press.

Barták, R., and Maillard, A. 2017. Attribute grammars with set attributes and global constraints as a unifying framework for planning domain models. In *Proc. of the 19th Int. Symp. on Principles and Practice of Declarative Programming (PPDP 2017)*, 39–48. ACM.

Barták, R.; Maillard, A.; and Cardoso, R. C. 2018. Validation of hierarchical plans via parsing of attribute grammars. In *Proc. of the 28th Int. Conf. on Autom. Plan. and Sched. (ICAPS 2018)*. AAAI Press.

Behnke, G.; Höller, D.; and Biundo, S. 2015. On the complexity of HTN plan verification and its implications for plan recognition. In *Proc. of the 25th Int. Conf. on Autom. Plan. and Sched. (ICAPS 2015)*, 25–33. AAAI Press.

Behnke, G.; Höller, D.; and Biundo, S. 2018. totSAT – Totally-ordered hierarchical planning through SAT. In *Proc. of the 32nd AAAI Conf. on AI (AAAI 2018)*, 6110–6118. AAAI Press.

Behnke, G.; Höller, D.; and Biundo, S. 2019. Bringing order to chaos – A compact representation of partial order in SAT-based HTN planning. In *Proc. of the 33rd AAAI Conf. on AI (AAAI 2019)*. AAAI Press.

Bercher, P.; Behnke, G.; Höller, D.; and Biundo, S. 2017. An admissible HTN planning heuristic. In *Proc. of the 26th Int. Joint Conf. on AI (IJCAI 2017)*, 480–488. IJCAI.

Dix, J.; Kuter, U.; and Nau, D. 2003. Planning in answer set programming using ordered task decomposition. In *Proc. of the 26th Annual German Conf. on AI (KI 2003)*, 490–504. Springer.

Dvorak, F.; Bit-Monnot, A.; Ingrand, F.; and Ghallab, M. 2014. A flexible ANML actor and planner in robotics. In *Proc. of the 4th Work. on Plan. and Rob. (PlanRob 2014)*, 12–19.

Erol, K.; Hendler, J.; and Nau, D. 1994. UMCP: A sound and complete procedure for hierarchical task-network planning. In *Proc. of the 2nd Int. Conf. on AI Plan. Systems (AIPS)*, 249–254. AAAI Press.

Erol, K.; Hendler, J.; and Nau, D. 1996. Complexity results for HTN planning. *Annals of Mathematics and AI* 18(1):69–93.

Geier, T., and Bercher, P. 2011. On the decidability of HTN planning with task insertion. In *Proc. of the 22nd Int. Joint Conf. on AI (IJCAI 2011)*, 1955–1961. AAAI Press.

Helmert, M. 2006. The fast downward planning system. *Journal of Artificial Intelligence Research (JAIR)* 26:191–246.

Höller, D.; Behnke, G.; Bercher, P.; and Biundo, S. 2014. Language classification of hierarchical planning problems. In *Proc. of the 21st Europ. Conf. on AI (ECAI 2014)*, volume 263, 447–452. IOS Press.

Höller, D.; Bercher, P.; Behnke, G.; and Biundo, B. 2018. A generic method to guide HTN progression search with classical heuristics. In *Proc. of the 28th Int. Conf. on Autom. Plan. and Sched. (ICAPS 2018)*, 114–122. AAAI Press.

McDermott, D. 2000. The 1998 AI planning systems competition. *AI Magazine* 21(2):35–55.

Muise, C., and Lipovetzky, N. 2015. Unplannability IPC track. In *Proc. of the 2015 Works. on the IPC (WIPC 2015)*.

Nau, D.; Cao, Y.; Lotem, A.; and Munoz-Avila, H. 1999. SHOP: Simple hierarchical ordered planner. In *Proc. of the 16th Int. Joint Conf. on AI (IJCAI 1999)*, 968–973.

Nau, D.; Au, T.-C.; Ilghami, O.; Kuter, U.; Murdock, J.; Wu, D.; and Yaman, F. 2003. SHOP2: An HTN planning system. *Journal of Artificial Intelligence Research (JAIR)* 20:379–404.

Ramoul, A.; Pellier, D.; Fiorino, H.; and Pesty, S. 2017. Grounding of HTN planning domain. *International Journal on Artificial Intelligence Tools* 26(5):1–24.

Schreiber, D.; Balyo, T.; Pellier, D.; and Fiorino, H. 2019. Tree-REX: SAT-based tree exploration for efficient and high-quality HTN planning. In *Proc. of the 29th Int. Conf. on Autom. Plan. and Sched. (ICAPS 2019)*. AAAI Press.

Shivashankar, V.; Alford, R.; Kuter, U.; and Nau, D. S. 2013. Hierarchical goal networks and goal-driven autonomy: Going where AI planning meets goal reasoning. In *Goal Reasoning: Papers from the ACS Workshop*, 95–110.

Smith, D.; Frank, J.; and Cushing, W. 2008. The anml language. In *Proc. of the Work. on Knowledge Engineering for Plan. and Sched. (KEPS 2008)*.