Received: 1 November 2023 | Revised: 25 January 2024 | Accepted: 5 February 2024

DOI: 10.1002/aaai.12169

MEETING REPORT

The 2023 International Planning Competition

Ayal Taitler[1] | Ron Alford[2] | Joan Espasa[3] | Gregor Behnke[4] | Daniel Fišer[5] | Michael Gimelfarb[1] | Florian Pommerening[6] | Scott Sanner[1] | Enrico Scala[7] | Dominik Schreiber[8] | Javier Segovia-Aguas[9] | Jendrik Seipp[10]

[1]University of Toronto, Toronto, Ontario, Canada

[2]The MITRE Corporation, McLean, Virginia, USA

[3]University of St Andrews, St Andrews, UK

[4]University of Amsterdam, Amsterdam, The Netherlands

[5]Saarland University, Saarbrucken, Saarland, Germany

[6]University of Basel, Basel, Switzerland

[7]University of Brescia, Brescia, Italy

[8]Karlsruhe Institute of Technology, Karlsruhe, Baden-Württemberg, Germany

[9]Universitat Pompeu Fabra, Barcelona, Spain

[10]Linköping University, Linkoping, Östergötland, Sweden

Correspondence
Ayal Taitler, University of Toronto, 27 King's College Cir, Toronto, Ontario, M5S 1A1, Canada.
Email: ataitler@gmail.com

Funding information
Swedish Research Council; University Of Edinburgh; EPSRC, Grant/Award Number: EP/P020267/1; Natural Sciences and Engineering Research Council of Canada; National Academic Infrastructure for Supercomputing in Sweden (NAISS), Grant/Award Number: 2022-06725; European Research Council, Grant/Award Number: 882500; MITRE; Independent Research and Development Program, Grant/Award Numbers: FA8702-19-C-0001, W56KGU-18-D-0004, DFARS 252.227-7013

Abstract

In this article, we present an overview of the 2023 International Planning Competition. It featured five distinct tracks designed to assess cutting-edge methods and explore the frontiers of planning within these settings: the classical (deterministic) track, the numeric track, the Hierarchical Task Networks (HTN) track, the learning track, and the probabilistic and reinforcement learning track. Each of these tracks evaluated planning methodologies through one or more subtracks, with the goal of pushing the boundaries of current planner performance. To achieve this objective, the competition introduced a combination of well-established challenges and entirely novel ones. Within this article, each track offers an exploration of its historical context, justifies its relevance within the planning landscape, discusses emerging domains and trends, elucidates the evaluation methodology, and ultimately presents the results.

INTRODUCTION

Automated planning (Ghallab, Nau, and Traverso 2004) is dedicated to the development of systems that can reason, strategize, and make decisions to accomplish specific objectives or maximize rewards in a complex and uncertain environment. It encompasses both domain-dependent planning, which tailors its strategies to specific problem domains, and domain-independent planning, which seeks to develop generalizable intelligent systems and algorithms. At its core, automated planning involves the formulation of actions, their prerequisites, and effects on the world, enabling machines to effectively address real-world challenges. Since 1998, the International Planning Competition (IPC) (Coles et al. 2012; Long and Fox 2003; McDermott 2000; Vallati et al. 2015) has been organized on a biennial or triennial basis with the objectives of promoting the advancement and evaluation of planning methodologies and coordinating the creation of new, challenging benchmarks.

This article provides an overview of IPC-2023, which featured a record-breaking number of five tracks, illustrating the rich diversity within the planning research community. This installment of the IPC is a reflection of the ongoing evolution within the field of planning, encompassing long-established tracks like classical planning, while also introducing new ones such as a numeric, and a probabilistic and reinforcement learning track. These latter instances underscore the significance of well-established tracks coexisting with the emerging trends of collaboration with the machine learning and reinforcement learning communities alongside the necessity of having efficient numeric reasoning in planners.

Portfolio submissions have been a hotly debated topic in the IPC. Given that many planning techniques work well in some domains but encounter difficulties with others, distributing the available time between different components or selecting components based on instance features typically achieves higher performance than running only the individual components. There are two main criticisms of portfolios: one concerns the attribution of success (running a portfolio of components A and B can surpass component A, even if A does most of the heavy lifting in the portfolio); the other pertains to scientific insight (analyzing the success of a portfolio is more challenging than that of its components). Some planners can clearly be seen as portfolios, and some can be clearly seen as non-portfolios. However, there is a large gray zone in between with techniques that use multiple components in a more integrated way. For example, *LAMA* (Richter, Westphal, and Helmert 2011) is typically not regarded as a portfolio but runs multiple different search algorithms and heuristics. Any specific definition of portfolios would provide an incentive to develop techniques that circumvent this definition, whether they are useful or not. For instance, if loose coupling of planning techniques were to define a portfolio, search methods that interleave multiple techniques in a sophisticated way would not be classified as portfolios but would still benefit from running all of their components. We decided not to draw this line and allowed portfolio submissions.

The five IPC-2023 tracks were as follows: The **Classical Track** is the oldest track and centers on fully observable environments where actions are atomic and have deterministic effects. In 1998, the Planning Domain Definition Language (PDDL) (McDermott 2000) was established as the standard for this and other tracks, and it continues to be developed and used to this day. The **Numeric Track** concentrates on deterministic planning problems with numeric state variables, emphasizing quantitative aspects of planning. Numeric planning problems can be expressed in the 2.1 version of PDDL (Fox and Long 2002). The **HTN Track** is focused on planning tasks that involve hierarchical structures. This is the second iteration of this track (Behnke, Höller, and Bercher 2021). As in the first iteration, this year's competition also uses the Hierarchical Domain Definition Language HDDL (Höller et al. 2020). The **Learning Track** evaluates systems that learn domain-specific knowledge and use it to assist a general planner in solving unseen tasks from the same planning domain. This track also uses PDDL as planner input language (Fern, Khardon, and Tadepalli 2011). The **Probabilistic and RL Track** is an advancement from the earlier probabilistic tracks (Younes et al. 2005), encompassing probabilistic and reinforcement learning elements. In the 2011 competition, the Relational Dynamic Influence Diagram Language (RDDL) (Sanner 2010b) has been introduced as the standard language for this track. RDDL was developed within the context of the competition, and an extended version is used in this year's competition (Taitler et al. 2022).

In the following sections, we survey each of these tracks in turn. A table summarizing the metadata of the competition across the tracks is given in Table 1.

CLASSICAL TRACK

History and motivation

The classical track has been part of the IPC since the inaugural competition in 1998, and this is the tenth IPC featuring it. The classical track focuses on the core of the planning problem without any of the extensions covered by the other tracks.

In the first three IPCs, all planners competed on the same playing field, but since 2004, there are subtracks for *optimal* planners (that guarantee plans with minimal costs) and *satisficing* planners (that have no such guarantee). With optimal planners, the goal is to find optimal plans for as many tasks as possible within some given resource limitations, while a good satisficing planner should discover short plans as quickly as possible. Historically, the *satisficing* track focused mostly on plan quality and ignored solution time. In 2014, an *agile* track was added that focused on solution time and ignored plan

TABLE 1 Metadata of the competition, including the tracks how many editions each track previously held, when was the last time the track took place, how many entries were submitted to each track in the 2023 event, the input language, and a link to each of the tracks webpage for additional information.

Track	Ed.	Last	Entries	Lang.	Webpage
Classical	9	2018	65	PDDL	https://ipc2023-classical.github.io/
Numeric	0	–	6	PDDL2.1	https://ipc2023-numeric.github.io/
HTN	1	2020	11	HDDL	https://ipc2023-htn.github.io/
Learning	3	2014	6	PDDL	https://ipc2023-learning.github.io/
Probabilistic	6	2018	4	RDDL	https://ataitler.github.io/IPPC2023/

quality. In 2023, we again offered an optimal, a satisficing, and an agile subtrack.

Compared to previous iterations of the competition, we emphasized PDDL features that make modeling a planning task easier. That is, we tried to promote the idea that writing a PDDL specification of a domain should be made as easy as possible, and planning systems should be able to perform normalization and preprocessing of the input planning task automatically. Our domains make full use of conditional effects, negative preconditions, quantifiers, disjunctions, imply conditions, and negative goal conditions. While conditional effects, negative preconditions, and to some degree quantifiers occur in previous IPC domains, the use of disjunctions and negative goal conditions is new and some planners did not support it yet. Our hope is that by introducing these features, planner authors will be pushed to add support for them. As an intermediate solution, we offered a tool to compile these features away. We also ran all planners on both the original domain, and one with those features compiled away, counting the better result in each domain.

New domains introduced

The classical track used seven new domains. In their selection, we focused on getting practically interesting domains without having particular planning techniques in mind, that is, the encoding or structure of the tasks is not intentionally optimized such that a particular technique works well.

Three domains, *Folding*, *Ricochet Robots*, and *Slitherlink*, were based on the domains previously used in Answer Set Programming competitions (e.g., Gebser, Maratea, and Ricca 2016, 2020). The reformulations of these domains used here contained zero-cost actions, disjunctions over static predicates, and negative goal conditions. The variant of the *Rubik's Cube* domain (modeling a well-known puzzle) used this year was formulated with a large number of conditional effects that are impossible to compile away without a significant blow-up of the planning tasks. Another domain, *Recharging Robots*, models a cooperative task of multiple robots that need to exchange a battery charge with each other in order to achieve the common goal. This domain was formulated with disjunctive preconditions, universal quantifiers, imply conditions, and conditional effects, all of which are possible to compile away without a significant blow-up of the final representation.

The *Labyrinth* domain was based on a board game where the agent has to travel through a maze that changes between the agent's movements. This domain contains auxiliary zero-cost actions, and it can be challenging for the grounding process as the maze can be arbitrarily reconfigured and therefore the agent can eventually move between almost all pairs of locations in one step. Finally, the winner of the Outstanding Domain Submission Award, *Quantum Circuit Layout Synthesis* (Shaik and van de Pol 2023), models a real-world problem of mapping a logical quantum circuit expressing a quantum algorithm to a real-world quantum circuit hardware platform. The tasks from this domain seem to be relatively easy to solve suboptimally, but getting optimal (or near optimal) solutions remains challenging.

All domains, except one, come with a task generator providing an opportunity to generate a different set of tasks possibly with a different scaling than the one used in the IPC. The only exception is the *Quantum Circuit Layout Synthesis* domain, which comes with PDDL formulations of the problems from the standard benchmark set used in the area of layout synthesis for quantum computing (Tan and Cong 2020). Moreover, each domain (again, except *Quantum Circuit Layout Synthesis*) is released with either a solver, or its generator provides at least a suboptimal solution alongside the output task.

Evaluation methodology

We used a submission framework very similar to the previous classical tracks at IPC 2018. Participants registered by email and we created repositories for them. They then pushed planner sources together with recipe scripts for the container solution Apptainer (Kurtzer, Sochat, and Bauer 2017). This way of handling submissions was tested in 2018, and has proven useful for several reasons: first, using repositories simplifies access management and code updates for possible bug fixes. Second, using a container solution simplified the build process on our side, in particular, dealing with a diverse set of library dependencies of the different planners. It also unifies the build setup and makes reusing the planners in a reproducible environment much easier.

Compared to 2018, we changed our bug-fixing policy: where in 2018, much effort was spent in identifying and catching bugs in the planners, we tried to outsource this effort to the planner authors who are more familiar with the behavior of their code. To that end, we had several rounds of test runs planners were evaluated on a subset of the competition instances. The results of these test runs and our parsed results where then published to the planner authors. They were then responsible for checking whether their planner performed as expected and our scripts interpreted the results correctly. If this was not the case, the authors could send pull requests with bug fixes both to their planners and to our scripts. We reviewed these pull request to ensure they did not tune the behavior of the planner to the known instances. This was an experimental step towards a potential full automation of the IPC similar to the Grid-Based Path Planning Competition[1].

After compiling the planner images, each image was run on all instances of a track for 30 min (optimal and satisficing tracks) or 5 min (agile track) with 8 GiB of available memory. In the optimal track, the score of an instance was 1 if it was solved optimally, and 0 otherwise. In the satisficing track, the score of a solved task is the ratio C^*/C where C is the cost of the cheapest discovered plan and C^* is the cost of a reference plan. The score on an unsolved task is 0. It is important that the reference plan is independent of the participants (Seipp 2019). We found optimal reference plans for most instances and plans that are at least as good as any plan found by a participant for most of the remaining cases. In very few cases, we had to fall back to using the best plan discovered by a participant as the reference solution. In the agile track, the score of a task is 1 if the task was solved within 1 s and 0 if the task was not solved within the resource limits. If the task was solved in T seconds ($1 \leq T \leq 300$) then its score is $1 - \log(T)/\log(300)$. This scoring accounts for the exponential growth in task difficulty while ignoring small differences in the subsecond range. In all tracks, each task, thus, gets a score between 0 and 1. These scores are summed up to achieve the final score of a planner. All domains have the same number of tasks, so normalizing with the domain size is not necessary.

After the competition, all domains, tasks, reference plans, and bounds were made available both through our homepage as well as the general planning task collection http://planning.domains. All planners are available as repositories on GitHub including their Apptainer recipies. They also can be installed through planutils (Muise et al. 2022). Planner abstracts for all planners are also available through the competition homepage. They include post-competition analyses where the authors analyze their performance retroactively.

Competition results and discussion

A total of 65 entries made by 23 teams participated across all classical tracks. Out of the 65 entries 25 self-identify as portfolios but as discussed earlier, the definition is not clear cut and other entries that use fall-back components for unsupported features could also classify as portfolios. We accepted multiple submissions of a planner if the authors suspected a strong difference in performance between two variants of their planner. Additionally, most planners participated in multiple subtracks. Thus, these 65 entries are based on 24 code bases. As the line between a single planner being configured differently for two subtracks and two different planners that share the same code base is blurry, we asked planner authors to submit separate planner abstracts if they consider two of their entries to be different planners. According to this metric, 34 distinct planners participated in the classical tracks. The teams consisted of 47 distinct authors from 19 affiliations where many authors contributed to more than one entry. We were happy to see that roughly half of the authors participated in an IPC for the first time.

As in the previous iterations, the classical planners submitted to the IPC 2023 utilized a wide range of techniques stretching to almost all corners of the classical planning area: From explicit state-space search to symbolic search to decoupled search, from partial order reduction to symmetry breaking to dominance pruning, from abstraction heuristics to delete-relaxation heuristics to heuristics based on linear programming. Competitors submitted lifted planners, ground planners, as well as portfolios of various techniques, equipped with several different grounders.

Overall, we think that IPCs have become a showcase of the wealth of classical planning techniques. This also means the final results should not be read and interpreted in a way where the winners represent the current state of the art and other entries do not (anymore). All techniques implemented in the submitted planners have their strengths and weaknesses and should be considered based on the structure of the task at hand. Moreover, the set of competition domains selected this year (or any other year) is by no means a representative of all possible tasks, and so the overall scores of the competing planners are necessarily biased. As discussed before, we tried to steer this bias in the direction of practically interesting domains rather than problems that are easily solved by

TABLE 2 Coverage for the optimal classical track. Best results per domain highlighted in gray.

	Ragnarok	*Scorpion 2023*	*Odin*	*Dofri*	*CEGAR++*	*Hapori Stone Soup*	*FD Stone Soup 2023*	*Hapori MIPlan*	*Hapori IBaCop2*	*Hapori Greedy*	*Baseline (blind)*	*DecStar 2023*	*Hapori Delfi*	*ComplementaryPDB*	*DecAbStar*	*Symk*	*FTSPlan (ms)*	*Baseline (LM-cut)*	*Hapori epslr*	*SymBD*	*QDom-Lmcut*	*Hapori epsdt*	*DALAI 2023*	*FTSPlan (sbd)*
folding	8	8	8	8	9	7	7	7	6	5	7	6	5	5	2	3	1	2	2	2	2	1	2	1
labyrinth	8	5	5	5	5	2	3	1	1	1	1	1	2	1	1	1	1	1	1	1	1	0	1	0
quantum.	13	14	13	13	13	13	13	13	13	13	7	12	12	12	12	9	12	12	9	9	12	9	11	4
recharg.	14	14	14	13	14	14	13	14	12	11	12	11	12	13	10	13	13	8	6	13	6	6	7	13
ricochet.	17	17	17	17	17	9	12	9	15	9	11	8	8	3	7	4	2	5	4	1	4	4	4	0
rubiks	10	10	10	10	0	11	9	10	7	10	8	8	0	0	0	0	0	0	10	0	0	7	0	0
slither.	7	6	6	4	7	6	4	6	4	7	4	4	2	3	5	7	7	6	2	6	6	4	4	4
Sum	77	74	73	70	65	62	61	60	58	56	50	50	41	37	37	37	36	34	34	32	31	31	29	22

a particular technique. Nevertheless, the IPC should encourage competitors to invest in their implementations, it should nurture healthy level of competitiveness as well as collaboration between researchers in the community, and above all it should be fun.

In the optimal track (see Table 2), 11 out of 22 competitors have a poor (or completely lack) support for conditional effects, which incurred a serious penalty in the overall scores because one of the domains (*Rubik's cube*) can not be solved without it. The structural variety among the selected domains seemed to be able to successfully probe various strengths and weaknesses of different planning techniques: Even planners in the second half of the leaderboard were able to achieve the highest (or close to the highest) score in some of the individual domains. The performance in domains *Folding*, *Labyrinth*, and *Ricochet Robots* seems to be what differentiate the top-ranking planners from the rest. The winner of the optimal track, *Ragnarok* (Drexler et al. 2023), is a portfolio planner of an explicit state-space search, decoupled search, symbolic search, and a lifted planner with various heuristics. It solved the highest number of tasks in four out of seven domains and only one below the highest score in the rest.

In the satisficing and agile track (see Tables 3 and 4), the planners performing best in individual domains were spread more evenly across the leaderboard than in the optimal track. Also, the support for conditional effects was more common than in the optimal track. The winners of the satisficing track, *Scorpion Maidu* (Corrêa et al. 2023b) and *Levitron* (Corrêa et al. 2023a), scored highest in only two domains. Both are portfolio planners that differ in that *Levitron* incorporates a lifted planner whereas *Scorpion Maidu* does not. The *DecStar* (Gnad, Álvaro Torralba, and Shleyfman 2023) planner was the winner of the agile track which, surprisingly, did not perform best in any individual domain but did well across all domains, especially in *Rubik's Cube*.

Our baseline planner, *LAMA* (winner of IPC 2011) (Richter, Westphal, and Helmert 2011), surprisingly achieved very high scores in the agile and satisficing track. No planner scored higher in the agile track and only three competing planners scored higher in the satisficing track. However, as it was considered a baseline, not a participant, it was not considered when determining the tracks' winners.

Overall, the biggest challenge across the board seemed to be the support for PDDL features such as quantifiers, disjunctions, imply conditions, and negative goal conditions. Since for every domain, we provided an alternative (automatically generated) formulation without these features, planners were not penalized in their final scores for not supporting them. However, it was often the case that planners performed better on the alternative formulation than on the original one. The most striking case was the *Slitherlink* domain, which uses just negative preconditions over static predicates (which is usually not problematic for current planners) and negative goal conditions. Yet, very few planners were able to solve any task in the original formulation of this domain, but they did much better in the alternative formulation where the negative goal conditions were compiled away with our tool.

NUMERIC TRACK

History and motivation

Planning with resources and numeric state variables in general has always been acknowledged as one of the most important features to support. Numeric state variables and

TABLE 3 Scores for the satisficing classical track. Best results per domain highlighted in gray. Hapori Greedy scored 0 due to a bug and is not shown.

	Scorpion Maidu	*Levitron*	*FD Stone Soup 2023*	*Baseline (LAMA)*	*Hapori IBaCop2*	*DiSCO*	*Cerberus*	*Spock*	*DecStar 2023*	*TFTM1*	*TFTM2*	*Hapori MIPlan*	*DALAI 2023*	*Apx Novelty*	*Hapori Stone Soup*	*FB Novelty Search*	*OpCount4Sat*	*Hapori epsdtl*	*Hapori Delfi*	*FSM*	*Powerlifted*	*Hapori epslr*
folding	6.8	8.7	9.0	9.7	8.7	8.6	6.4	5.8	5.4	6.7	6.7	5.6	4.8	5.0	7.7	1.0	2.0	7.4	3.8	3.0	9.7	1.0
labyrinth	0.0	0.0	0.0	1.0	3.9	0.0	0.0	0.0	0.0	0.0	0.0	3.9	4.0	0.0	3.9	0.0	3.8	14.1	0.0	0.0	0.0	0.0
quantum.	19.6	19.6	19.5	17.9	17.0	17.8	18.0	19.7	17.3	16.7	15.8	18.9	17.3	18.3	18.4	18.3	13.3	0.0	12.5	6.6	16.7	0.0
recharg.	13.9	13.9	13.8	13.2	13.5	14.0	10.4	13.9	13.3	9.6	10.4	13.3	13.0	8.0	12.3	8.0	10.1	5.4	8.9	8.2	0.0	3.5
ricochet.	11.4	11.4	8.5	9.8	7.4	9.1	12.7	5.0	8.4	12.7	12.7	5.0	8.8	15.3	4.2	13.8	6.2	0.6	2.4	3.0	0.0	0.0
rubiks	14.2	14.2	14.1	12.2	11.2	11.8	10.9	13.9	12.4	10.0	10.1	10.0	5.0	5.0	4.0	5.0	0.0	6.0	4.0	6.0	0.0	11.0
slither.	6.0	4.0	6.0	5.0	6.0	5.0	6.0	6.0	6.0	6.0	6.0	4.0	3.0	4.0	4.0	4.0	4.0	0.0	1.0	3.0	2.0	5.0
Sum	71.9	71.8	70.9	68.8	67.7	66.2	64.5	64.4	62.9	61.7	61.7	60.7	56.0	55.6	54.5	50.1	39.3	33.6	32.5	29.8	28.4	20.5

TABLE 4 Scores for the agile classical track. Best results per domain highlighted in gray. Hapori Greedy scored 0 due to a bug and is not shown.

	Baseline (LAMA)	*DecStar 2023*	*FD Stone Soup 2023*	*Apx Novelty Tarski*	*DiSCO*	*Scorpion Maidu*	*Levitron*	*Apx Novelty*	*Cerberus*	*DALAI 2023*	*Hapori Stone Soup*	*Hapori MIPlan*	*TFTM1*	*TFTM2*	*FB Novelty Search*	*FTSPlan*	*Hapori epsdtl*	*Hapori epslr*	*Hapori Delfi*	*OpCount4Sat*	*FSM*	*Powerlifted*
folding	3.4	2.6	3.0	1.9	3.4	2.5	2.5	1.4	2.7	3.0	0.9	0.4	2.7	2.6	0.1	0.0	1.8	0.0	0.7	0.0	0.0	0.7
labyrinth	0.0	0.0	1.0	2.7	0.0	0.3	0.3	0.0	0.0	0.3	0.7	0.8	0.0	0.0	0.0	0.0	2.7	0.0	0.0	0.0	0.0	0.0
quantum.	16.9	15.3	15.3	19.5	16.8	15.6	15.6	19.6	13.5	13.6	16.7	13.6	12.2	12.3	18.9	9.6	0.0	0.0	5.0	3.0	0.8	0.0
recharg.	3.8	5.0	4.5	3.9	6.5	4.4	4.1	3.0	2.4	6.6	3.9	2.5	2.5	2.4	0.8	5.2	2.8	2.5	2.4	6.3	1.0	0.0
ricochet.	0.9	2.7	0.6	3.4	1.1	1.6	1.0	3.4	1.8	3.5	0.8	0.8	1.7	1.9	1.2	0.6	0.5	0.0	1.5	0.0	0.2	0.0
rubiks	13.2	13.0	11.0	3.1	6.3	9.6	9.5	4.0	7.8	2.7	3.9	7.1	6.0	6.0	4.0	0.0	5.2	7.5	0.6	0.0	3.2	0.0
slither.	2.2	1.7	2.4	1.8	1.9	1.9	2.6	1.8	2.6	1.0	1.6	2.7	2.6	2.6	0.5	2.0	0.0	1.0	0.5	0.7	0.0	0.0
Sum	40.3	40.3	37.8	36.4	36.0	35.9	35.6	33.3	30.8	30.8	28.5	27.8	27.7	27.7	25.5	17.3	13.0	10.9	10.8	10.1	5.3	0.8

relative constraints represent a very compact device to succinctly encode domains involving coordinates, resources such as fuel or energy and every exotic thing one can think of when it is necessary to reason quantitatively over some aspect of the world the agent is operating. Yet, so far, there has never been a competition that was specifically designed to study the behavior of planning systems that have to deal with numeric reasoning. Indeed, since its first introduction, numeric planning has appeared in the competition only together with temporal planning (Long and Fox 2003). This track has been motivated by a revived interest in this specific setting (e.g., Bofill, Espasa, and Villaret 2019; Kuroiwa et al. 2022; Leofante et al. 2020; Piacentini et al. 2018; Scala et al. 2020) and therefore, the need to focus attention over numeric reasoning in planners, keeping other aspects such as temporal aspects aside.

Numeric planning, in a formal sense, does not represent a substantial departure from classical planning. Both types of planning still rely on a deterministic transition system to model the world. However, there is a key distinction: while classical planning problems typically induce finite transition systems, numeric planning problems can result in infinite state spaces. This characteristic makes numeric planning problems undecidable in general, as discussed by Helmert (2002); decidability can be recovered only if variables are properly bounded (Gigante and Scala 2023).

Numeric planning allows the definition of numeric conditions (e.g., $(x + y \geq 10)$) and effects (e.g., $(x := 10 + z)$). Albeit numeric planning can in its general form support nonlinear expressions both in the conditions and in the effects, to lower the participation barrier, we only consider two simpler fragments: *simple numeric planning*

(SNP) (Scala et al. 2020) and *linear numeric planning* (LNP). SNP is a simpler subset of numeric planning, where numeric variables can only be increased or decreased by a constant and numeric conditions are limited to linear expressions. This subset is expressive enough to allow the formalization of interesting problems. LNP extends SNP by not only allowing numeric variables to be assigned directly, but also allowing increases, decreases, and assignments to linear combinations of variables.

Preconditions and goals in the evaluated problems were arbitrary propositional formulas. That is, they included quantifiers, negative preconditions, disjunctive preconditions, and any number of numeric predicates. Conditional effects were not considered. Terms were either literals as in classical planning, or numeric terms of the form $f(X)\ \{\geq, >, =\}\ 0$ where $f(X)$ is a linear expression.

Similarly to the classical track, the numeric track also featured three different subtracks: *optimal*, where plans with minimal costs are guaranteed, *agile*, which only focused on solution time and ignored plan quality, and *satisficing*, which aimed to find the best possible plans in the given time limit.

In the optimal track, the optimization functions were always a minimization. Problems were either a minimization of action costs, or a metric function that was a weighted sum with positive coefficients where each involved variable could only be increased by the actions.

New domains introduced

Given that resources and numeric state variables are a natural extension from classical planning problems, unless the domains were chosen carefully, there was the risk of evaluating numeric planners on solving mainly Boolean structures. Therefore, there was an emphasis on selecting domains that exhibited a notable numeric structure.

The origins of the domains used for the competition were a mix of domains used in previous IPCs (such as *Zenotravel* or *Settlers* from the IPC-5), interesting domains introduced in various publications over the years (such as *FO-Sailing* or *Counters*) and some new domains (such as *Drone* or *Ext-Plant-Watering*). A total of 20 domains were selected to be used in the competition, of which seven were LNP domains and 13 SNP domains. In turn, each domain had 20 instances selected with various degrees of difficulty.

Evaluation methodology

We collected a grand total of six planners from two different teams.

TABLE 5 Winners of the Numeric Planning Track by subtrack.

Track	Winner
Opt SNP	*NLM-CutPlan Orbit*
Opt LNP	*NLM-CutPlan*
Opt SNP+LNP	*LM-CutPlan Orbit*
Sat SNP	*NLM-CutPlan SAT2*
Sat LNP	*NLM-CutPlan SAT*
Sat SNP+LNP	*NLM-CutPlan SAT*
Agile SNP	*NLM-CutPlan SAT2*
Agile LNP	*NLM-CutPlan SAT*
Agile SNP+LNP	*NLM-CutPlan SAT*

The first team submitted the *NLM-Plan* system, which exposed four different heuristic search planners. These four planners are described in detail in an extended abstract[2]. The second team submitted *OMTPlan*, which exposed two different SMT-based planners. Those are described in detail in a recent publication (Leofante 2023). All participants have been given a GitHub account that they used to upload the source code of the planners. We then compiled all the source code and run all experiments in the Cirrus HPC Service at EPCC. Each run used a Intel Xeon E5-2695 (Broadwell) with a limit of 8-GiB memory limit per process. The competition used exactly the same scoring formulas as the classical planning track. All planners and benchmarks are available at https://github.com/ipc2023-numeric.

Competition results and discussion

As not all submitted planners supported the LNP fragment, we had different rankings for SNP, LNP, and for both together. The NLM-based planners won all subtracks. More precisely: *NLM-CutPlan* won the Optimal LNP, *NLM-CutPlan SAT2* won the Satisficing SNP and the Agile SNP subtracks, *NLM-CutPlan Orbit* won the two other *optimal* tracks, that is, Optimal SNP and Optimal SNP + LNP, and *NLM-CutPlan SAT* won all the remaining tracks, as summarized by Table 5.

As baselines, we used a well-known greedy best first search algorithm with state of the art heuristics from the literature (Scala et al. 2020) and A^* with the numeric extension of the h_{rmax} heuristic (Scala et al. 2017) for the satisficing and the optimal track, respectively. These are implemented in the ENHSP planning system[3]. The competing planners were not able to improve the baseline used in the satisficing track, showing the strength of the current heuristic there. On the other hand, *NLM-Plan* notably improved the capacity of numeric planners on proving optimality. Finally, considering the competitors, no planner had total dominance: *OMTPlan* was the best planner in 17% of the instances. When considering the number of SNP + LNP instances solved per domain, *OMTPlan* tied with *NLM-Plan* in five domains and won in two domains.

We envision a number of challenges for the community. Where, in fact, there has been a substantial advancement in the optimal setting, for the satisficing setting the baseline proved to be even more competitive than the participanting planners. By a closer look at the raw data, we can, however, observe that this gap is only pronounced in SNP problems. For linear planning problems, and in particular over the linear version of the *Sailing* domain, the baseline solved zero instances, while *NLM-CutPlan Sat* solved up to the 14th instance. Therefore, it is possible that much more can be done for problems requiring more advanced forms of reasoning, for instance, those involving nonlinear dependencies. Another aspect worth investigation is portfolio solutions. Indeed, as observed before, no planner had total dominance. Properly characterizing numeric planning problems would allow the creation of portfolios, opening the possibility of combining the best approaches into one single planner; this approach has already proven to be useful into classical planning, and is likely that the same applies to the numeric context, too.

HTN TRACK

History and motivation

Hierarchical Task Network (HTN) planning allows automated planning formalisms to be extended with complex hierarchical expert knowledge, imposing additional structural constraints for valid plans (Bercher, Alford, and Höller 2019). A planning task is represented not only by a set of actions and an initial state but also a set of *tasks* to be achieved. Primitive tasks can be achieved by executing them as actions, whereas compound tasks are achieved by recursively replacing them with *subtasks*, following certain conditional decomposition methods. In *Total Order* (TO) HTN planning, all sets of subtasks as well as the set of initial tasks are totally ordered, that is, they are sequences. This restriction is popular and relevant since it renders the formalism decidable, as opposed to semi-decidable *Partial Order* (PO) HTN planning where arbitrary ordering constraints can be imposed between tasks (Erol, Hendler, and Nau 1994). The high expressive power of HTN planning is exploited by many applications, for example, cooperative robotics (Bevacqua et al. 2015), AI in video games (Vellido, Fdez-Olivares, and Pérez 2020), or assistance for complex handicraft tasks (Behnke et al. 2019).

IPC-2023 featured the second iteration of an HTN planning track, following IPC-2020, which focused entirely on HTN planning. As in 2020, all benchmark problems were provided in the HDDL format (Höller et al. 2020), in its adapted version for the IPC 2020.[4] We selected a total of 22 TO planning domains and 10 PO domains for evaluating planners. While the majority of planning domains were already a part of IPC-2020 (namely 20 TO domains and eight PO domains), four new domains were introduced.

#SAT (a.k.a. *SharpSAT*, TO domain) models a simple algorithm (Birnbaum and Lozinskii 1999) for *exact model counting*, that is, counting the number of satisfying assignments for a given propositional formula. This task is #ℙ-complete and thus a challenging task even for very small propositional formulas. *Lamps* (TO) models a variant of the game “Lights Out,” which is about an $n \times m$ field with lamps that can either be on or off. Switching a lamp forces all horizontally and vertically connected lamps of the same status (on or off) to also toggle. This reachability-based procedure can be easily modeled with an HTN, but is hard to express using classical planning. *Ultralight-Cockpit* (PO) models emergency procedures in ultralight aircraft. Given the current situation of the aircraft and the nature of the emergency, the plan represents instructions to the pilot on how to handle this emergency and how to recover the aircraft's situation safely. *Coloring* (PO) encodes a version of the tiling problem (van Emde Boas 1997), which is frequently used for complexity reductions. Given a set of available tiles, each having a color at one of its edges, the task is to fill an $n \times n$ square with these tiles, s.t., touching edges have the same color. The outer edge has no required color. This problem is ℕℙ-complete for unary encoded n. The encoding uses the idea of proof encoding double-exponentially time-bounded Turing Machine (Alford, Bercher, and Aha 2015).

Evaluation methodology

We offered two times three subtracks: TO and PO HTN planning on the one hand, and agile, satisficing and optimal scoring as in the classical and numeric tracks on the other hand. We measure plan length in terms of the number of actions, that is, primitive tasks, that are executed in the hierarchical solution.

Each submission was allowed up to three configurations per track. Planning approaches submitted by the same group were considered different submissions if the approaches' inner workings are sufficiently different. Planners were allowed to use up to 8 GiB of RAM and 30 min of running time per instance.

As in the classical track, participants had to notify the organizers via email of their intent and were then provided with a git repository for their planner. The HTN track also used Apptainer to automate compiling and running the planners. After the feature-stop, we performed test runs of the planners on the IPC domains. The organizers then searched for anomalies in the runs and notified the participants, who were given the opportunity to propose a pull-request to fix these issues, which were checked and accepted by the organizers.

Competition results and discussion

We received a total of eleven distinct submissions. Only a single submission, *SIADEX* (Fernandez-Olivares, Vellido, and Castillo 2021), participated in IPC-2020 as well. IPC-2020 TO winner *HyperTensioN* (Magnaguagno, Meneguzzi, and de Silva 2021) and runner-up *Lilotane* (Schreiber 2021) did not participate in 2023 (although two 2023 submissions, *Lifted-Linear* and *LTP*, build upon *Lilotane*). Some HTN planning approaches did not compete because they are (co-)authored by some of the organizers (Alford et al. 2016; Behnke 2021; Behnke et al. 2022; Behnke and Speck 2021; Schreiber 2021).

The submitted planners feature a good variety of different algorithms and techniques. *SIADEX* (I. V. Expósito, J. Soler-Padial, J. F. Olivares, L. Castillo), *PandaDealer* (C. Olz, D. Höller, P. Bercher), *PANDApro* (D. Höller), and *PANDA λ* (D. Höller) are progression search planners. *Aries* (A. Bit-Monnot), *LTP* (*Lifted Tree Path*) (G. Quenard, D. Pellier, H. Fiorino), and *OptiPlan* (O. Firsov, D. Pellier, H. Fiorino) perform planning via constraint programming and/or SAT solving. *TOAD* Höller (2021) uses a translation to classical planning. Lastly, the submissions *Lifted-Linear*, *Linear-Simple*, and *Linear-Complex* (Y. X. Wu, C. Olz, S. Lin, P. Bercher) perform a preprocessing which linearizes PO into TO problems and then use one of several established TOHTN planners.

The submissions are roughly balanced in terms of ground versus lifted planning: *Linear-Simple*, *Linear-Complex*, *OptiPlan*, *PandaDealer*, *PANDApro*, and *PANDA λ* use a grounding procedure to simplify and prune the problem before planning. *Aries*, *SIADEX*, *LTP*, and *Lifted-Linear* do not perform grounding but rather operate directly on the parametrized description.

Table 6 shows the top performing systems for each subtrack. Across all TO tracks, *PandaDealer* was the most successful planner. It was followed rather closely by *PANDA λ* (agile, satisficing) and *PANDApro* (optimal). *PandaDealer* builds upon *PANDApro* and features a look-ahead technique, which evaluates and prunes branches early by checking state-related conditions that are propagated from preconditions and effects of primitive tasks (Olz et al. 2021). In the PO tracks, *Linear-Simple* and *Linear-Complex* dominated the satisficing and agile subtracks. Their winning configurations use ground progression search via *PANDA*$_\pi$ or *PandaDealer* after linearizing the problem into a TO problem—falling back to regular POHTN planning in cases where the linearization proves to be unsolvable. *PANDApro* and *PANDA λ* were the best approaches, which performed direct POHTN planning.

TABLE 6 Winners and runner-ups of the HTN track. In case of multiple top rankings of a single system, only the 1–2 best configurations are shown.

Track		Winner	Runner-up
TO	Sat	*PandaDealer-agile-lama*	*PANDA λ ao*
	Agl	*PandaDealer-agile-{1,lama}*	*PANDA λ lm-cut*
	Opt	*PandaDealer-optimal*	*PANDApro {lm-cut,dof}*
PO	Sat	*Linear-Complex-satisficing-1* *PANDA λ {ao,lm-cut}*	
	Agl	*Linear-Simple-agile-2*	*PANDA λ {ao,lm-cut}*
	Opt	*PANDApro lm-cut*	*Aries*

Since planners behave differently across domains, it is important to also consider domain-dependent results. For example, in the TO satisficing track, the best configuration of *TOAD* was able to score the best in seven domains—two more than the winner (a *PandaDealer* configuration). It, however, performed poorly in several domains (including three without any solved instance)—*TOAD* had better peak performance, but was less reliable than *PandaDealer*. Let us also consider a virtual planner *V* which, for each TO domain, selects the planner which solved most instances. Overall *V* solves 590 instances—45 more than any single planner (*PandaDealer*)—and, at least, features *PandaDealer*, *SIADEX*, *LTP*, and *TOAD*.

The IPC 2020 was dominated by lifted planners—both winners and the runner-up of the TO track were lifted (*HyperTensioN*, *SIADEX*, and *Lilotane*). In both tracks, there was a significant performance gap between the lifted and grounded planners. Out of this year's winners and runner-ups, only *Aries* (runner-up PO optimal) is lifted. Lifted planning did prove to remain worthwhile especially in the TO agile track: Four domains were solved best by *SIADEX* alone and three domains were solved best by *LTP* alone. All in all, the results, however, emphasize the progress brought by the latest progression search approaches with efficient and effective grounding (Behnke et al. 2020), new pruning strategies (Höller and Behnke 2021), and informed search decisions (Höller et al. 2019). These approaches outperform lifted and translation-based approaches on the majority of domains but do not fully dominate them. In terms of POHTN planning, this year's results suggest that many of the considered PO problems can be solved effectively by TOHTN planners through careful transformation. It remains to be seen whether such compilations continue to gain traction in future research to accelerate POHTN planning.

LEARNING TRACK

History and motivation

The learning track was introduced 15 years ago and 2023 saw its fourth appearance at the IPC. The track consists of two separate phases: one for *learning* some type of domain knowledge given a set of training instances from a planning domain; and the evaluation phase, in which a *solver* uses the learned domain knowledge to solve unseen instances from the same domain.

In the previous editions (2008, 2011, and 2014), the domains, example instances and instance generators for the training phase were made available to the participants, so they could learn the domain knowledge on their machines. Afterwards, the participants submitted the learned domain knowledge and their solvers to the organizers, who ran them on the evaluation tasks. Although the aim was to learn domain knowledge that allows to scale up to larger and more complex instances, usually the instances used in the first and second phase were of similar difficulty.

In this year's edition, we made three fundamental changes to the learning track: first, we fully automated the learning phase and let participants submit a learner that computes domain knowledge for a set of approximately 100 easy instances per domain without user intervention. Second, we evaluated the second part of their submission, a planner that uses the domain knowledge, on 30 easy, 30 medium, and 30 hard instances from the same domain. Third, we extended the traditional STRIPS fragment of the Planning Domain Definition Language (PDDL) with typed objects and negative preconditions.

Our main motivation for these changes is to (1) increase reproducibility of research on the topic of learning for planning, where currently it is very hard to set up fair comparisons to baseline learning approaches, (2) encourage research on learning scalable domain knowledge instead of simply learning how to tweak the configuration of a planning system for a given domain, and (3) to motivate the community to support a richer task input language.

The IPC 2014 edition of the Learning Track featured a "Best Learner Award" given to the system that maximizes the difference in plan quality between system versions running without and with the learned domain knowledge. In contrast, we only consider the latter, because for many systems, it is not clear what would constitute a nonlearning version and how to ensure it has reasonable performance.

Domains

The competition used ten classical planning domains from earlier IPCs: *Blocksworld*, *Childsnack*, *Ferry*, *Floortile*, *Miconic*, *Rovers*, *Satellite*, *Sokoban*, *Spanner*, and *Transport*. Their hardness ranges from being optimally solvable in polynomial time (e.g., *Spanner*), to PSPACE-complete domains like *Sokoban* (Culberson 1997). To obtain reference plans for the evaluation tasks, we developed domain-dependent solvers for all ten domains and validated all reference plans using the Unified Planning framework.[5] All benchmarks, domain-dependent solvers, and reference plans are available via the competition website, as well as all code submitted by the participants.

Evaluation methodology

We originally envisioned both a single-core CPU-only environment and a multi-core environment with GPU access. Since only one planner, *Muninn*, opted for the latter, we canceled the GPU variant. Thankfully, the *Muninn* authors agreed to compete in the CPU environment even though this heavily disfavors their approach. We limited the time and memory for computing domain knowledge to 24 h and 32 GiB, respectively. For the evaluation phase, we allowed at most 30 min and 8 GiB of memory per task. To evaluate the submissions, we used the same quality and agile scores as in the classical tracks. Since there were 10 domains with 90 evaluation tasks each, the maximum score was 900 points. Both the learners and the solvers were allowed to generate improved domain knowledge and plans as they progressed. Since the two metrics yielded the same ranking, we only report the quality score in the following.

Competition results and discussion

A total of six teams, each with one submission, participated in the learning track: *ASNets* 2023 (Hao et al. 2023) ported Action Schema Networks (Toyer et al. 2020) to Tensorflow 2. This neural network architecture can encode a generalized reactive policy by learning a common set

TABLE 7 Plan quality scores for the Learning Track. For the missing entries, no domain knowledge was learned.

	Baselines			Competitors				
	LAMA	*FDSS*	*SMAC*	*ASNets*	*GOFAI*	*HUZAR*	*Muninn*	*Vanir*
Blocksworld	47.9	49.4	31.5	4.6	46.4	39.3	40.6	–
Childsnack	26.2	35.4	20.2	0.0	26.5	22.0	11.0	–
Ferry	64.0	61.5	64.4	–	58.5	58.7	42.1	76.3
Floortile	12.0	22.7	24.7	–	34.4	21.3	0.0	–
Miconic	84.4	89.6	52.3	7.2	81.4	72.4	30.0	75.2
Rovers	66.8	64.0	58.1	6.5	54.4	60.0	14.2	66.1
Satellite	87.3	88.7	71.0	–	74.0	79.9	16.0	87.3
Sokoban	37.7	39.0	30.8	0.0	38.4	28.1	24.3	37.7
Spanner	30.0	60.7	30.0	8.9	30.0	30.0	32.0	–
Transport	61.4	63.0	62.7	2.0	64.5	55.4	16.2	–
Sum	517.6	574.1	445.7	29.1	508.5	467.0	226.3	342.6

of weights for all tasks in a domain. *GOFAI* (Torralba and Gnad 2023) learns which action schema instantiations are likely part of a plan. Then it uses this information to *partially* ground the given task (Gnad et al. 2019) and solve it with a Fast Downward configuration optimized by SMAC (Hutter, Hoos, and Leyton-Brown 2011). *HUZAR* (Gzubicki, Lachowicz, and Torralba 2023) learns to distinguish between good and bad transitions by feeding problem description graphs into a graph neural network. *Muninn* (Ståhlberg, Bonet, and Geffner 2023) uses relational message-passing neural networks (Ståhlberg, Bonet, and Geffner 2022) to learn a value function and follows it first in a hill-climbing search and then (if unsuccessful) within an A* search. *NPGP* (Lei, Lipovetzky, and Ehinger 2023b) is a novelty-based generalized planner (Lei, Lipovetzky, and Ehinger 2023a) that prunes a newly generated planning program (Segovia-Aguas et al. 2022) if its most frequent action repetition is greater than a given bound. *Vanir* (Drexler 2023) learns width-based hierarchical policies for polynomial domains (Drexler, Seipp, and Geffner 2023).

Table 7 shows the quality scores of the submitted systems. We omit *NPGP* from the table because it failed to learn domain knowledge (DK) in the evaluation domains due to the inherent limitations of the *PGP* learner it builds upon, that is, it cannot handle domains with hierarchical typing nor action schemas using constants.[6] *ASNets 2023* also experienced difficulties, failing to learn DK in three domains and producing weak DK in seven others, receiving a quality score of just 29.1 points. Even though *Muninn* is optimized for GPUs, it was able to learn DK in all domains and even surpassed all other competitors in the *Spanner* domain. However, in total, it obtained only a score of 226.3 points.

Vanir, which specifically targets polynomial domains, only produced domain knowledge files for five domains, but the quality was high, achieving the highest score among competitors in three domains (*Ferry*, *Rovers*, and *Satellite*). Overall, *Vanir* achieved an overall score of 342.6 points.[7] *HUZAR* is the runner-up winner of the competition with an overall quality score of 467.0 points, even though it stops after finding the first plan.

The competition was won by *GOFAI* (508.5 points) with its anytime approach to iteratively obtaining cheaper plans. *GOFAI* managed to achieve scores as high or higher than the *Fast Downward SMAC* baseline that participated in the IPC 2014 Learning Track in eight out of 10 domains, indicating significant progress since the last iteration of the competition 9 years ago.

Despite the strides, the competition highlighted key areas for improvement, especially since *GOFAI* scored lower than the nonlearning baselines *LAMA 2023* (Richter and Westphal 2010) and *FDSS 2023* (Büchner et al. 2023) in five and eight domains, respectively. Future challenges are to develop robust learning systems that can handle more PDDL features, new domains and task distributions, to create learning algorithms that scale to harder domains, and planning systems using DK that outperform domain-independent classical planners on very large instances.

PROBABILISTIC AND RL TRACK

History and motivation

The probabilistic and RL track of IPC-2023 represents a significant departure from the probabilistic tracks held under the IPC umbrella in previous years. Since 2011, the probabilistic track presented problems described using the Relational Dynamic Influence Diagram Language (RDDL) Sanner (2010b). This year, the track continued this tradition but introduced a single track with problems described in an extended version of RDDL. While previous iterations of the track focused on discrete MDPs and PODMPs, this year, the emphasis shifted entirely to continuous and mixed discrete-continuous MDPs, including a single industry-contributed discrete problem among the eight problems featured in the track.

There were two main goals in this year's track: (Goal 1) This goal aimed to redirect the community's focus toward realistic problems that showcase state exogenous noise, the effects of stochastic actions, concurrency, and critically *mixed discrete-continuous transition dynamics*. (Goal 2) Motivated by the observation that both the reinforcement learning and planning communities are interested in the types of domains outlined in the first goal, the second goal aimed to unite the probabilistic planning and reinforcement learning communities under a common competition and software framework. This aim also resulted in the name change of the track.

To enable the seamless integration of both reinforcement learning (RL) and planning methods, the original Java-based RDDLSim (Sanner 2010a) simulation framework has been replaced with a Python-based framework called *pyRDDLGym* (Taitler et al. 2022) that includes a fast vectorized simulator. This framework serves as an auto-generation toolkit for OpenAI Gym environments (Brockman et al. 2016), directly generated from raw RDDL files. As a result, *pyRDDLGym is fully compatible with the standard interaction model and framework of the RL community*. Additionally, it accommodates model-based planning methods by providing the RDDL domain and instance files that describe the model along with compilations both to Jax computation graphs (Bradbury et al. 2018) as well as an extended Algebraic Decision Diagram (XADD) (Sanner, Delgado, and de Barros 2011) format to represent the dynamic Bayesian network and influence graphs for the factored MDP underlying a ground RDDL instance.

JaxPlan, a model-based back-propagation planner, Taitler et al. (2022), is included out-of-the-box with *pyRDDLGym*. It is provided to assist competitors in getting started, and used as an evaluation baseline, as explained in the Evaluation Methodology section.

Participants in the current year's probabilistic and RL track engaged in competitions spanning eight distinct domains. The track achieved an all-time high in registration numbers, with 29 competitors showing interest and signing up during the initial phase. In the second stage, which took place a month and a half prior to the competition, all participants were tasked with submitting a working planner. This stage served as a dry-run to assess the infrastructure and procedures, resulting in only four competitors successfully advancing to the competition.

New domains introduced

As part of the comprehensive overhaul of this year's track, all eight domains offered were entirely new. The problems exhibited a diverse range of properties; some were goal-oriented, while others involved steady-state control. One notable challenge this year was an oversubscribed multi-agent version of the *Mars-Rover* problem, inspired by (Yliniemi, Agogino, and Tumer 2014). An interesting addition was the *RecSim* domain—a Recommender Systems domain contributed by Google (Mladenov et al. 2020). Although it was the only discrete domain in the competition, it featured a vast number of enumerable state and action spaces, reaching up to 40,000 actions per time step in the largest instance of the competition.

Furthermore, a *UAV* problem with multiple drones was introduced based on a simplified version of the dynamics described in Hull (2007). The *UAV* problem incorporated both controllable and uncontrollable dynamic-model UAVs, with uncontrollable UAVs not contributing to the reward. This aspect called for model reasoning to determine which parts of the space were worth ignoring.

The domains were also categorized by problem type, ranging from classical control problems like *Mountain-Car* and *RaceCar* to operations research (OR) problems such as *Reservoir* control and *Power Generation*, and even extending to e-commerce.

Evaluation methodology

The online phase of this track took place in early June 2023, spanning a week. At the beginning of the week, three instances of each domain, gradually increasing in difficulty, were released to competitors. These instances were generated using prereleased instance generators, and their generation parameters were also provided. Competitors had 1 week to fine-tune their methods, and at the week's end, they submitted their planners. These submitted planners were evaluated on the three instances that had already been released, as well as on two additional instances that had not been seen before. The difficulty level of these new instances was designed to be more challenging than the easiest instance but less difficult than the hardest. The objective was to maximize the reward within a finite time horizon.

The scores for each instance were precisely normalized within the [0,1] range. Any episode's reward that fell below the maximum reward achievable through a random policy or a no-operation policy (default unperturbed behavior) was set to zero. Conversely, the method with the highest accumulated reward, including the organizers' baseline planner, set the upper boundary at 1, and all other scores were normalized accordingly. *JaxPlan* served as a baseline and was executed in two modes, functioning as two separate baselines: straight-line planning (SLP) mode and deep reactive policy (DRP) mode (Bueno et al. 2019). The average reward for each instance was computed as the mean of 50 independent trials. Each trial had a four-minute allocation, with an additional 60 min provided before the trials for automatic hyper-parameter tuning.

The overall winner of the competition was the method that achieved the highest accumulated score across all instances, totaling five instances over eight domains for a total of 40 instances. Consequently, the score ranged from 0 to 40.

TABLE 8 Summary of results for the Probabilistic Track for the two baselines *JaxPlan-SLP* and *JaxPlan-DRP* and the winning planner *DiSProD*. Each entry counts the number of tasks won per domain.

Domain	*SLP*	*DRP*	*DiSProD*
HVAC	1	0	4
MarsRover	0	4	1
MountainCar	4	1	3
UAV	0	0	5
PowerGen	1	2	2
Reservoir	3	2	0
RaceCar	5	0	0
RecSim	1	1	3
Total	15	10	**18**

Competition results and discussion

Four teams successfully qualified for the online stage of the competition by meeting all the requirements, which included submitting an abstract and passing a dry-run evaluation. These approaches were categorized into one planning approach and three model-free RL approaches. One of the RL teams and the planning team represented academia, while the other two RL teams were from industry.

Only the planning approach managed to complete a successful final submission that could be evaluated. This winning team of *DiSProD* (P. Chatterjee, A. Chapagain, R. Khardon) built upon their previous *DiSProD* work (Chatterjee et al. 2023), a gradient-based search approach capable of propagating distributions between time steps. To accommodate the time constraints of the competition (4 min per trial), a more time-efficient variant was employed.

This year's competition provided a fascinating perspective on the field of probabilistic planning and learning, particularly in terms of the problem types offered. The initial registration numbers set an all-time record, and the contributions of domains and code from the community suggest a strong interest within both the research and industry communities in the types of challenges presented in this year's competition. However, the significant drop in the number of competitors at each stage of the competition, and the fact that only one team successfully crossed the finish line, with that team opting for a planning approach, adds complexity to the overall picture.

The most apparent challenges include the necessity for methods capable of solving both continuous and discrete problems and the requirement to tackle tasks encompassing control, operations research (OR), navigation, and more, without domain-specific adjustments. We would like to emphasize two notable challenges:

Instance size: One of the major factors affecting the difficulty level of the instances is the size of the state and action spaces. While differentiable planning approaches such as *DiSProD* scale relatively well versus instance size (and are mostly limited by memory constraints), RL methods are significantly impacted by instance size since increases in the number of state and action variables of the underlying MDP typically lead to challenges with effective exploration and an explosion of the sample complexity (and therefore learning time) required to obtain a reasonably performing policy.

Exploiting structure: This year's track placed considerable emphasis on problems with inherent sparse transition structure that can be leveraged for efficiency. For example, consider an *HVAC* problem with one heater and two rooms. In the first instance, the rooms can be isolated, while in the next, they can be adjacent and allow for heat transfer. Model-free RL methods have no knowledge of this underlying transition structure and are unable to exploit any knowledge of the independence that occurs in the first instance. In contrast, model-based planning methodologies are able to exploit this independence, for example, *DiSProD*'s gradient-based optimization method would exploit the fact that one room's temperature does not depend on the other room since such a dependence would not arise in the calculated derivatives (Table 8).

In summary, this year's track drew participants from academia and industry, showcasing interest in complex challenges. The competition highlighted the need for versatile methods to handle diverse tasks, ultimately favoring a differentiation-based planning approach. Integrating model-based planning and model-free RL approaches remains a challenge in this evolving field. Equally importantly, we were not aware that any methods attempted to leverage generalized planning and RL approaches (Sanner and Boutilier 2009; Sharma et al. 2023) that could exploit domain structure to generalize learning over instances, which we see as a key to improving the efficiency of

RL methods while providing broad generalization abilities to new instances with little (or no) need for per-instance adaptation.

CONCLUDING REMARKS

As in previous iterations, the International Planning Competition has been instrumental in advancing the field of automated planning, bringing together researchers from various domains to push the boundaries of planning methodologies. IPC-2023 was notable for its diverse set of tracks, reflecting the diversity within the planning research community.

The classical track continued to evolve, emphasizing features to facilitate modeling while maintaining optimal, satisficing, and agile subtracks. The numeric track explored the capabilities of state-of-the-art planners in dealing with numeric reasoning; an effective numeric reasoning is a crucial building block to handle planning models that are closer to reality. The HTN track highlighted the significance of efficient grounding and domain-specific approaches in handling complex hierarchical tasks. The learning track addressed the fully automatic acquisition and application of domain-specific knowledge, showcasing advancements while revealing areas for further development. Lastly, the probabilistic and reinforcement learning track introduced new complexities and challenges to the field, paving the way for hybrid methods, combining tools from the planning and learning communities.

IPC-2023 demonstrates the advances within the planning community, embracing new trends, and coexisting with traditional methodologies. With recent advances in AI blurring the historical barriers between AI fields, future planning competitions might consider opportunities for the cross-pollination and integration of planning in novel application-focused tracks to further showcase the capabilities of planning methodologies in the presently burgeoning AI ecosystem.

ACKNOWLEDGMENTS

Some of the computations for the Learning track were enabled by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council through Grant Agreement No. 2022-06725. The numeric track computations used the Cirrus UK National Tier-2 HPC Service at EPCC (http://www.cirrus.ac.uk) funded by the University Of Edinburgh and EPSRC (EP/P020267/1). The HTN track has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Grant Agreement No. 882500). This work was partially funded as part of the MITRE Independent Research and Development Program. Portions of this technical data were produced for the U. S. Government under Contract No. FA8702-19-C-0001 and W56KGU-18-D-0004, and is subject to the Rights in Technical Data-Noncommercial Items Clause DFARS 252.227-7013 (FEB 2014). The Probabilistic and Reinforcement Learning Track has received funding form the Natural Sciences and Engineering Research Council of Canada (NSERC).

CONFLICT OF INTEREST STATEMENT

The authors declare that there is no conflict.

ORCID

Ayal Taitler https://orcid.org/0000-0003-3919-6883
Ron Alford https://orcid.org/0000-0002-3688-0658
Joan Espasa https://orcid.org/0000-0002-9021-3047

ENDNOTES

[1] https://gppc.search-conference.org/

[2] https://ipc2023-numeric.github.io/abstracts/NLM_CutPlan_Abstract.pdf

[3] https://gitlab.com/enricos83/ENHSP-Public/-/tags/enhsp20-0.9.9

[4] Clarifications and changes are explained in https://github.com/panda-planner-dev/ipc2020-domains/blob/master/hddl.pdf.

[5] https://github.com/aiplan4eu/unified-planning

[6] These and other issues are being addressed in the *BFGP++* solver (https://github.com/jsego/bfgp-pp), an extended version of *PGP*.

[7] Due to a bug, *Vanir* failed to run its backup planner (*LAMA*) when no hierarchical policy could be learned.

REFERENCES

Alford, R., G. Behnke, D. Höller, P. Bercher, S. Biundo, and D. W. Aha. 2016. "Bound to Plan: Exploiting Classical Heuristics via Automatic Translations of Tail-Recursive HTN Problems." In *Proceedings of the 26th International Conference on Automated Planning and Scheduling, (ICAPS 2016)*, 20–28. AAAI Press.

Alford, R., P. Bercher, and D. W. Aha. 2015. "Tight Bounds for HTN Planning." In *Proceedings of the 25th International Conference on Automated Planning and Scheduling (ICAPS 2015)*, 7–15. AAAI Press.

Behnke, G. 2021. "Block Compression and Invariant Pruning for SAT-Based Totally-Ordered HTN Planning." In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 31, 25–35.

Behnke, G., D. Höller, and P. Bercher, eds. 2021. *Proceedings of the 10th International Planning Competition: Planner and Domain Abstracts – Hierarchical Task Network (HTN) Planning Track (IPC 2020)*. https://ipc2020.hierarchical-task.net/publications/IPC2020Booklet.pdf.

Behnke, G., D. Höller, A. Schmid, P. Bercher, and S. Biundo. 2020. "On Succinct Groundings of HTN Planning Problems." In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 9775–84.

Behnke, G., F. Pollitt, D. Höller, P. Bercher, and R. Alford. 2022. "Making Translations to Classical Planning Competitive with Other HTN Planners." In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI 2022)*, 9687–97. AAAI Press.

Behnke, G., M. Schiller, M. Kraus, P. Bercher, M. Schmautz, M. Dorna, M. Dambier, W. Minker, B. Glimm, and S. Biundo. 2019. "Alice in DIY-Wonderland Or: Instructing Novice Users on How to Use Tools in DIY Projects." *AI Communications* 32(1): 31–57.

Behnke, G., and D. Speck. 2021. "Symbolic Search for Optimal Total-Order HTN Planning." In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 11744–54.

Bercher, P., R. Alford, D. Höller. 2019. "A Survey on Hierarchical Planning – One Abstract Idea, Many Concrete Realizations." In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI 2019)*, ed. S. Kraus, 6267–75. IJCAI.

Bevacqua, G., J. Cacace, A. Finzi, and V. Lippiello. 2015. "Mixed-Initiative Planning and Execution for Multiple Drones in Search and Rescue Missions." In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 25, 315–23.

Birnbaum, E., and E. L. Lozinskii. 1999. "The Good Old Davis-Putnam Procedure Helps Counting Models." *Journal of Artificial Intelligence Research* 10: 457–77.

Bofill, M., J. Espasa, and M. Villaret 2019. "Relaxing Non-Interference Requirements in Parallel Plans." *Logic Journal of the IGPL* 29(1): 45–71.

Bradbury, J., R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, et al. 2018. "JAX: Composable Transformations of Python+NumPy Programs." http://github.com/google/jax.

Brockman, G., V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. 2016. "Openai Gym." *arXiv preprint arXiv:1606.01540*.

Büchner, C., R. Christen, A. B. Correêa, S. Eriksson, P. Ferber, J. Seipp, and S. Sievers. 2023. "Fast Downward Stone Soup 2023." In *Tenth International Planning Competition (IPC-10): Planner Abstracts*.

Bueno, T. P., L. N. de Barros, D. D. Mauá, and S. Sanner. 2019. "Deep Reactive Policies for Planning in Stochastic Nonlinear Domains." In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 7530–37.

Chatterjee, P., A. Chapagain, W. Chen, and R. Khardon. 2023. "DiSProD: Differentiable Symbolic Propagation of Distributions for Planning." *in Proceddings of the Thirty-Second International Joint Conference on Artificial Intelligence, 5324-2332, IJCAI.*

Coles, A., A. Coles, A. G. Olaya, S. Jiménez, C. L. López, S. Sanner, and S. Yoon. 2012. "A Survey of the Seventh International Planning Competition." *AI Magazine* 33(1): 83–88.

Corrêa, A. B., G. Francès, M. Hecher, D. M. Longo, and J. Seipp. 2023a. "Levitron: Combining Ground and Lifted Planning." In *Tenth International Planning Competition (IPC-10): Planner Abstracts*.

Corrêa, A. B., G. Francès, M. Hecher, D. M. Longo, and J. Seipp. 2023b. "Scorpion Maidu: Width Search in the Scorpion Planning System." In *Tenth International Planning Competition (IPC-10): Planner Abstracts*.

Culberson, J. C. 1997. "Sokoban is PSPACE-Complete." Technical Report TR 97-02, Department of Computing Science, The University of Alberta, Edmonton, Alberta, Canada.

Drexler, D. 2023. "Vanir: Learning and Executing Width-based Hierarchical Policies." In *Tenth International Planning Competition (IPC-10) Learning Track: Planner Abstracts*.

Drexler, D., D. Gnad, P. Höft, J. Seipp, D. Speck, and S. Ståhlberg. 2023. "Ragnarok." In *Tenth International Planning Competition (IPC-10): Planner Abstracts*.

Drexler, D., J. Seipp, and H. Geffner. 2023. "Learning Hierarchical Policies by Iteratively Reducing the Width of Sketch Rules." In *Proceedings of the Twentieth International Conference on Principles of Knowledge Representation and Reasoning (KR 2023)*.

Erol, K., J. Hendler, and D. S. Nau. 1994. "HTN Planning: Complexity and Expressivity." In *AAAI*, volume 94, 1123–28.

Fern, A., R. Khardon, and P. Tadepalli. 2011. "The First Learning Track of the International Planning Competition." *Machine Learning* 84: 81–107.

Fernandez-Olivares, J., I. Vellido, and L. Castillo. 2021. "Addressing HTN Planning with Blind Depth First Search." In *Proceedings of 10th International Planning Competition: Planner and Domain Abstracts – Hierarchical Task Network (HTN) Planning Track (IPC 2020)*, 1–4.

Fox, M., and D. Long. 2002. "The Third International Planning Competition: Temporal and Metric Planning." In *AIPS*, 333.

Gebser, M., M. Maratea, and F. Ricca. 2016. "What's Hot in the Answer Set Programming Competition." In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, eds. D. Schuurmans and M. P. Wellman, 4327–29. AAAI Press.

Gebser, M., M. Maratea, and F. Ricca. 2020. "The Seventh Answer Set Programming Competition: Design and Results." *Theory and Practice of Logic Programming* 20(2): 176–204.

Ghallab, M., D. Nau, and P. Traverso. 2004. *Automated Planning: Theory and Practice*. Amsterdam: Elsevier.

Gigante, N., and E. Scala. 2023. "On the Compilability of Bounded Numeric Planning." In *IJCAI*, 5341–49. ijcai.org.

Gnad, D., Á. Torralba, and A. Shleyfman. 2023. "DecStar." In *Tenth International Planning Competition (IPC-10): Planner Abstracts*.

Gnad, D., Á. Torralba, M. A. Domínguez, C. Areces, and F. Bustos. 2019. "Learning How to Ground a Plan – Partial Grounding in Classical Planning." In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI 2019)*, 7602–9. AAAI Press.

Gzubicki, P. R., B. P. Lachowicz, and Á. Torralba. 2023. "HUZAR: Predicting Useful Actions with Graph Neural Networks." In *Tenth International Planning Competition (IPC-10) Learning Track: Planner Abstracts*.

Hao, M., S. Toyer, R. Wang, S. Thiébaux, and F. Trevizan. 2023. "Action Schema Networks – IPC Version." In *Tenth International Planning Competition (IPC-10) Learning Track: Planner Abstracts*.

Helmert, M. 2002. "Decidability and Undecidability Results for Planning with Numerical State Variables." In *Proceedings of the International Conference on Artificial Intelligence Planning and Scheduling (AIPS 2002)*, 44–53.

Höller, D. 2021. "Translating Totally Ordered HTN Planning Problems to Classical Planning Problems Using Regular Approximation of Context-Free Languages." In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 31, 159–67.

Höller, D., and G. Behnke. 2021. "Loop Detection in the PANDA Planning System." In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 31, 168–73.

Höller, D., G. Behnke, P. Bercher, S. Biundo, H. Fiorino, D. Pellier, and R. Alford. 2020. "HDDL: An Extension to PDDL for Expressing Hierarchical Planning Problems." In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 9883–91.

Höller, D., P. Bercher, G. Behnke, and S. Biundo. 2019. "On Guiding Search in HTN Planning with Classical Planning Heuristics." In *IJCAI*, 6171–75.

Hull, D. G. 2007. *Fundamentals of Airplane Flight Mechanics*, volume 19. Berlin: Springer.

Hutter, F., H. Hoos, and K. Leyton-Brown. 2011. "Sequential Model-Based Optimization for General Algorithm Configuration." In *Proceedings of the Fifth Conference on Learning and Intelligent OptimizatioN (LION 2011)*, ed. C. A. C. Coello, 507–23. Springer.

Kuroiwa, R., A. Shleyfman, C. Piacentini, M. P. Castro, and J. C. Beck. 2022. "The LM-Cut Heuristic Family for Optimal Numeric Planning with Simple Conditions." *Journal of Artificial Intelligence Research* 75: 1477–548.

Kurtzer, G. M., V. Sochat, and M. W. Bauer. 2017. "Singularity: Scientific Containers for Mobility of Compute." *PLOS ONE* 12(5): 1–20.

Lei, C., N. Lipovetzky, and K. Ehinger. 2023a. "Novelty and Lifted Helpful Actions in Generalized Planning." In *Proceedings of the 16th Annual Symposium on Combinatorial Search (SoCS 2023)*, eds. R. Barták, W. Ruml, and O. Salzman, 148–52. AAAI Press.

Lei, C., N. Lipovetzky, and K. A. Ehinger. 2023b. "IPC Learning Track: Novelty-Based Generalized Planning." In *Tenth International Planning Competition (IPC-10) Learning Track: Planner Abstracts*.

Leofante, F. 2023. "OMTPlan: A Tool for Optimal Planning Modulo Theories." *Journal on Satisfiability, Boolean Modeling and Computation* 14(1): 17–23.

Leofante, F., E. Giunchiglia, E.Ábrahám, and A. Tacchella. 2020. "Optimal Planning Modulo Theories." In *IJCAI*, 4128–34. ijcai.org.

Long, D., and M. Fox. 2003. "The 3rd International Planning Competition: Results And Analysis." *Journal of Artificial Intelligence Research* 20: 1–59.

Magnaguagno, M. C., F. Meneguzzi, and L. de Silva. 2021. "HyperTensioN: A Three-Stage Compiler for Planning." In *Proceedings of 10th International Planning Competition: Planner and Domain Abstracts – Hierarchical Task Network (HTN) Planning Track (IPC 2020)*, 5–8.

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

Mladenov, M., E. Creager, O. Ben-Porat, K. Swersky, R. Zemel, and C. Boutilier. 2020. "Optimizing Long-Term Social Welfare in Recommender Systems: A Constrained Matching Approach." In *International Conference on Machine Learning*, 6987–98. PMLR.

Muise, C., F. Pommerening, J. Seipp, and M. Katz. 2022. "Planutils: Bringing Planning to the Masses." In *ICAPS 2022 System Demonstrations and Exhibits*.

Olz, C., E. Wierzba, P. Bercher, and F. Lindner. 2021. "Towards Improving the Comprehension of HTN Planning Domains by Means of Preconditions and Effects of Compound Tasks." In *Proceedings of the 10th Workshop on Knowledge Engineering for Planning and Scheduling, KEPS*.

Piacentini, C., M. P. Castro, A. A. Ciré, and J. C. Beck. 2018. "Linear and Integer Programming-Based Heuristics for Cost-Optimal Numeric Planning." In *AAAI*, 6254–61. AAAI Press.

Richter, S., and M. Westphal. 2010. "The LAMA Planner: Guiding Cost-Based Anytime Planning with Landmarks." *Journal of Artificial Intelligence Research* 39: 127–77.

Richter, S., M. Westphal, and M. Helmert. 2011. "LAMA 2008 and 2011 (planner abstract)." In *IPC 2011 Planner Abstracts*, 50–54.

Sanner, S. 2010a. "RDDLSim." https://github.com/ssanner/rddlsim.

Sanner, S. 2010b. "Relational Dynamic Influence Diagram Language (RDDL): Language Description." Unpublished manuscript. Australian National University 32: 27.

Sanner, S., and C. Boutilier. 2009. "Practical Solution Techniques for First-order MDPs." *Artificial Intelligence Journal (AIJ)* 748–88. Recipient of the 2014 Artificial Intelligence Journal (AIJ) Prominent Paper Award.

Sanner, S., K. V. Delgado, and L. N. de Barros. 2011. "Symbolic Dynamic Programming for Discrete and Continuous State MDPs." In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence (UAI-11)*. Barcelona, Spain.

Scala, E., P. Haslum, D. Magazzeni, and S. Thiébaux. 2017. "Landmarks for Numeric Planning Problems." In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI*, ed. C. Sierra, 4384–90. ijcai.org.

Scala, E., P. Haslum, S. Thiébaux, and M. Ramirez. 2020. "Subgoaling Techniques for Satisficing and Optimal Numeric Planning." *Journal of Artificial Intelligence Research* 68: 691–752.

Schreiber, D. 2021. "Lilotane: A Lifted SAT-based Approach to Hierarchical Planning." *Journal of Artificial Intelligence Research* 70: 1117–81.

Segovia-Aguas, J., S. J. Celorrio, L. Sebastiá, and A. Jonsson. 2022. "Scaling-Up Generalized Planning as Heuristic Search with Landmarks." In *Proceedings of the 15th Annual Symposium on Combinatorial Search (SoCS 2022)*, eds. L. Chrpa and A. Saetti, 171–79. AAAI Press.

Seipp, J. 2019. "Planner Metrics Should Satisfy Independence of Irrelevant Alternatives." In *ICAPS 2019 Workshop on the International Planning Competition (WIPC)*, 40–41.

Shaik, I., and J. van de Pol. 2023. "Optimal Layout Synthesis for Quantum Circuits as Classical Planning." *in Proceedings of the IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, 1–9, IEEE.

Sharma, V., D. Arora, Mausam, and P. Singla. 2023. "SymNet 3.0: Exploiting Long-Range Influences in Learning Generalized Neural Policies for Relational MDPs." In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, eds. R. J. Evans and I. Shpitser, volume 216 of *Proceedings of Machine Learning Research*, 1921–31. PMLR.

Ståhlberg, S., B. Bonet, and H. Geffner. 2022. "Learning Generalized Policies without Supervision Using GNNs." In *Proceedings of the Nineteenth International Conference on Principles of Knowledge Representation and Reasoning (KR 2022)*, 474–83.

Ståhlberg, S., B. Bonet, and H. Geffner. 2023. "Muninn." In *Tenth International Planning Competition (IPC-10) Learning Track: Planner Abstracts*.

Taitler, A., M. Gimelfarb, S. Gopalakrishnan, M. Mladenov, X. Liu, and S. Sanner. 2023. "pyRDDLGym: From RDDL to Gym Environments." *In ICAPS 2023 Workshop on Bridging the Gap Between AI Planning and Reinforcement Learning (PRL)*.

Tan, B., and J. Cong. 2020. "Optimal Layout Synthesis for Quantum Computing." In *IEEE/ACM International Conference on Computer Aided Design, ICCAD 2020, San Diego, CA, USA, November 2-5, 2020*, 1–9. IEEE.

Torralba, Á., and D. Gnad. 2023. "GOFAI." In *Tenth International Planning Competition (IPC-10) Learning Track: Planner Abstracts*.

Toyer, S., S. Thiébaux, F. Trevizan, and L. Xie. 2020. "ASNets: Deep Learning for Generalised Planning." *Journal of Artificial Intelligence Research* 68: 1–68.

Vallati, M., L. Chrpa, M. Grześ, T. L. McCluskey, M. Roberts, S. Sanner. 2015. "The 2014 International Planning Competition: Progress And Trends." *AI Magazine* 36(3): 90–98.

van Emde Boas, P. 1997. "The Convenience of Tilings." *in Complexity, Logic and Recursion Theory*, by Sorbi, Andrea, Boca Raton, CRC Press, 331–63.

Vellido, I., J. Fdez-Olivares, and R. Pérez. 2020. "A Knowledge Based Process for the Generation of HTN Domains from VGDL Video Game Descriptions." In *ICAPS2020 Workshop on Knowledge Engineering for Planning and Scheduling*, 11–20.

Yliniemi, L., A. K. Agogino, and K. Tumer. 2014. "Multirobot Coordination for Space Exploration." *AI Magazine* 35(4): 61–74.

Younes, H. L., M. L. Littman, D. Weissman, and J. Asmuth. 2005. "The First Probabilistic Track of the International Planning Competition." *Journal of Artificial Intelligence Research* 24: 851–87.

How to cite this article: Taitler, A., R. Alford, J. Espasa, G. Behnke, D. Fišer, M. Gimelfarb, F. Pommerening, S. Sanner, E. Scala, D. Schreiber, J. Segovia-Aguas, and J. Seipp 2024. "The 2023 International Planning Competition." *AI Magazine* 45: 280–96. https://doi.org/10.1002/aaai.12169

AUTHOR BIOGRAPHIES

Ayal Taitler is a Postdoctoral Research Fellow at the University of Toronto. He was a Co-organizer of the Probabilistic and RL Track.

Ron Alford is a Researcher at the MITRE Corporation. He was a Co-organizer of the HTN Track.

Joan Espasa is a Lecturer at the University of St Andrews. He was a Co-organizer of the Numeric Track.

Gregor Behnke is an Assistant Professor at the University of Amsterdam. He was a Co-organizer of the HTN Track.

Daniel Fišer is a Postdoctoral Researcher at Saarland University. He was a Co-organizer of the Classical Track.

Michael Gimelfarb is a Postdoctoral Research Fellow at the University of Toronto. He was a Co-organizer of the Probabilistic and RL Track.

Florian Pommerening is Postdoctoral Research Fellow at the University of Basel. He was a Co-organizer of the Classical Track.

Scott Sanner is an Associate Professor at the University of Toronto. He was a Co-organizer of the Probabilistic and RL Track.

Enrico Scala is a Senior Lecturer at the University of Brescia. He was a Co-organizer of the Numeric Track.

Dominik Schreiber is a Postdoctoral Researcher at Karlsruhe Institute of Technology. He was a Co-organizer of the HTN Track.

Javier Segovia-Aguas is a Postdoctoral Research Fellow at the Universitat Pompeu Fabra. He was a Co-organizer of the Learning Track.

Jendrik Seipp is an Associate Professor at Linköping University. He was a Co-organizer of the Learning Track.