Hierarchical Planning in the IPC

Gregor Behnke, Daniel Höller, Pascal Bercher, Susanne Biundo,
Damien Pellier, Humbert Fiorino,
Ron Alford

Ulm University, Germany
University Grenoble Alpes, France
The MITRE Corporation, USA

July 2019

ulm university universität
uulm

Just Why?

The IPC helped the planning community

Just Why?

The IPC helped the planning community

- to focus on a common problem

Just Why?

The IPC helped the planning community

- to focus on a common problem
- to develop common code-bases (e.g. FD)

Just Why?

The IPC helped the planning community

- to focus on a common problem
- to develop common code-bases (e.g. FD)
- to define common benchmark sets

Just Why?

The IPC helped the planning community

- to focus on a common problem
- to develop common code-bases (e.g. FD)
- to define common benchmark sets
- to define a common description language (PDDL, RDDL, ...)

Just Why?

The IPC helped the planning community

- to focus on a common problem
- to develop common code-bases (e.g. FD)
- to define common benchmark sets
- to define a common description language (PDDL, RDDL, ...)
- to make planners useable by a larger community

Just Why?

The IPC helped the planning community

- to focus on a common problem
- to develop common code-bases (e.g. FD)
- to define common benchmark sets
- to define a common description language (PDDL, RDDL, ...)
- to make planners useable by a larger community

The hierarchical planning (sub-)community

Just Why?

The IPC helped the planning community

- to focus on a common problem
- to develop common code-bases (e.g. FD)
- to define common benchmark sets
- to define a common description language (PDDL, RDDL, ...)
- to make planners useable by a larger community

The hierarchical planning (sub-)community

- has no common description language

Just Why?

The IPC helped the planning community

- to focus on a common problem
- to develop common code-bases (e.g. FD)
- to define common benchmark sets
- to define a common description language (PDDL, RDDL, ...)
- to make planners useable by a larger community

The hierarchical planning (sub-)community

- has no common description language
- has no consensus on features that must be supported

Just Why?

The IPC helped the planning community

- to focus on a common problem
- to develop common code-bases (e.g. FD)
- to define common benchmark sets
- to define a common description language (PDDL, RDDL, ...)
- to make planners useable by a larger community

The hierarchical planning (sub-)community

- has no common description language
- has no consensus on features that must be supported
- has no standard benchmarks to compare planners

Just Why?

The IPC helped the planning community

- to focus on a common problem
- to develop common code-bases (e.g. FD)
- to define common benchmark sets
- to define a common description language (PDDL, RDDL, ...)
- to make planners useable by a larger community

The hierarchical planning (sub-)community

- has no common description language
- has no consensus on features that must be supported
- has no standard benchmarks to compare planners
- often cannot compare planners for theoretical reasons

Just Why?

The IPC helped the planning community

- to focus on a common problem
- to develop common code-bases (e.g. FD)
- to define common benchmark sets
- to define a common description language (PDDL, RDDL, ...)
- to make planners useable by a larger community

The hierarchical planning (sub-)community

- has no common description language
- has no consensus on features that must be supported
- has no standard benchmarks to compare planners
- often cannot compare planners for theoretical reasons
- is generally lacking focus

Potential Competitor

General HTN planning

- FAPE a temporal HTN planner with strong pruning techniques
- PANDA a plan-space planner using heuristic search
- PANDApro a progression-based planning system using heuristic search
- HTN2STRIPS a planner translating HTN planning problems into a sequence of classical planning problems
- partSAT a planner based on a translation into propositional logic

Potential Competitor

General HTN planning

- FAPE a temporal HTN planner with strong pruning techniques
- PANDA a plan-space planner using heuristic search
- PANDApro a progression-based planning system using heuristic search
- HTN2STRIPS a planner translating HTN planning problems into a sequence of classical planning problems
- partSAT a planner based on a translation into propositional logic

Totally-Ordered HTN planning

- GTOHP a planner based on intelligent grounding and blind search
- totSAT planners that translates (totally-ordered) HTN planning problems into propositional logic
- HTN2ASP a planner that translates (totally-ordered) HTN planning problems into answer set programming

Potential Competitor

May organisers also compete?

Potential Competitor

May organisers also compete?

If the process is transparent.

- First submission of planners prior to submission of domains
- Any change after the first submission is public (git)

Potential Competitor

May organisers also compete?

If the process is transparent.

- First submission of planners prior to submission of domains
- Any change after the first submission is public (git)
- Maybe restrict allowed changes for organisers?

Tracks

- Optimal
- Satisficing
- Agile

- General HTNs without state constraints
- General HTNs with state constraints (# competitors?)
- Totally-Ordered HTNs

Setting

We propose to use standard IPC setting

- 1 CPU core
- 8 GB of RAM
- 30 minutes

Setting

We propose to use standard IPC setting

- 1 CPU core
- 8 GB of RAM
- 30 minutes

For Domains we propose a mix of

- Current benchmark domains (as far as compatible)
- Community submissions
- BYOB = Bring-Your-Own-Benchmark
  - Each submitted planner must provide a domain with 20 (?) instances
  - The submitting planner must solve at most half of these instances
  - Used by SAT-Races 2017 and 2018

Verification

Plans produced by the planner must be verified.

Verification

Plans produced by the planner must be verified.

- Verification of action sequences is $\mathbb{NP}$-complete for HTN planning problems

Verification

Plans produced by the planner must be verified.

- Verification of action sequences is $\mathbb{NP}$-complete for HTN planning problems
- $\mathbb{P}$ when planners output the decomposition (i.e. derivation)

Verification

Plans produced by the planner must be verified.

- Verification of action sequences is $\mathbb{NP}$-complete for HTN planning problems
- $\mathbb{P}$ when planners output the decomposition (i.e. derivation)
- Two possible approaches

Verification

Plans produced by the planner must be verified.

- Verification of action sequences is $\mathbb{NP}$-complete for HTN planning problems
- $\mathbb{P}$ when planners output the decomposition (i.e. derivation)
- Two possible approaches
  1. Use two existing HTN plan verifier (Barak et al. & Behnke et al.)

Verification

Plans produced by the planner must be verified.

- Verification of action sequences is $\mathbb{NP}$-complete for HTN planning problems
- $\mathbb{P}$ when planners output the decomposition (i.e. derivation)
- Two possible approaches
  1. Use two existing HTN plan verifier (Barak et al. & Behnke et al.)
  2. Require planners to output decomposition using a – yet to be defined – standard format

Scoring

- 50% of respondents in the questionnaire were ok with the IPC-score
- Other metrics were proposed

Scoring

- 50% of respondents in the questionnaire were ok with the IPC-score
- Other metrics were proposed
  - Number of backtracking operations

Scoring

- 50% of respondents in the questionnaire were ok with the IPC-score
- Other metrics were proposed
  - Number of backtracking operations
    Not all planners backtrack, it is highly algorithm-specific

Scoring

- 50% of respondents in the questionnaire were ok with the IPC-score
- Other metrics were proposed
  - Number of backtracking operations
    Not all planners backtrack, it is highly algorithm-specific
  - Depth of search

Scoring

- 50% of respondents in the questionnaire were ok with the IPC-score
- Other metrics were proposed
  - Number of backtracking operations
    Not all planners backtrack, it is highly algorithm-specific
  - Depth of search
    Not well defined

Scoring

- 50% of respondents in the questionnaire were ok with the IPC-score
- Other metrics were proposed
  - Number of backtracking operations
    Not all planners backtrack, it is highly algorithm-specific
  - Depth of search
    Not well defined
  - Number of methods

Scoring

- 50% of respondents in the questionnaire were ok with the IPC-score
- Other metrics were proposed
  - Number of backtracking operations
    Not all planners backtrack, it is highly algorithm-specific
  - Depth of search
    Not well defined
  - Number of methods
    All planners must use the same problem

Timetable

May – July 2019	Agreeing on a common input language
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