International Planning Competition

Milan Ježek
Outline

- IPC 2014 – overview
- Deterministic track
- Other tracks
- Challenge – what is my intention
- Possible approaches
IPC 2014 - Overview

• Nearly biennial event
• In the context of the ICAPS-14, Portsmouth (USA)

• Goals:
  • Empirical comparison of planning systems
  • Highlighting challenges to the community
  • New directions for research
  • New links with other fields of AI
  • New data sets for benchmarks
IPC 2014 – Important dates (for deterministic track)

- June 13: Call for Participation available
- June 13: Call for Domains available
- July 13: Competition Rules are now available
- Oct, 31, 13: Registration deadline
- Nov, 15, 13: Demo problems, supporting tools
- Nov, 15, 13: Domain submission deadline
- Jan, 17, 14: Planners submission deadline
- March, 28, 14: Papers submission deadline
Four different Tracks

- Deterministic Track
- Learning Track
- Probabilistic Planning Track
  - Continuous
- Probabilistic Planning Track
  - Discrete
Deterministic Track

- Three tracks
  - **Sequential** track
  - **Temporal** track
  - **Preferences** track
- Two different subtracks for each track
  - **Optimal, Satisficing** subtracks
- Additional two subtracks for **Sequential** track
  - **Multi-core, Agile** subtracks
Deterministic - **Sequential**

- Classical STRIPS planning (non-durative actions)
- Non-negative costs
- Negative preconditions and conditional effects
- Reasonable time, low-cost plans
- Core features: STRIPS, action costs, negative preconditions, conditional effects
- Optional features: ADL, derived predicates
- **Total cost** of each plan is the sum of the **costs of its actions**
- Objective function: minimize total cost
Deterministic - Temporal

- Temporal planning with **metric constraints**
- Core features: **STRIPS**, durative actions, **metric quantities**
- Optional features: ADL, derived predicates
- Objective function: **Minimize makespan**
Deterministic - Preferences

- Planning with **soft goals**
- Valid plan **does not have to achieve all goals**
- Not achieving a goal implies a certain **penalization** added to the cost of the plan
- Cost of the plan is a combination of the total actions cost and penalizations
- Core features: **STRIPS, action costs, goal utilities, metric quantities**
- Optional features: ADL, derived predicates
- Objective function: **Minimize total cost**
Optimal variants

- 30 minutes to solve each problem
- What matters is only whether the problem was solved or not
- Plans have to be optimal
- At least one plan in a given domain is non-optimal
  ➞ all results of that planner in that domain are ignored
- At least one non-optimal plan on at least two different domains
  ➞ the planner is disqualified
- Objective function: maximize number of solved problems
Satisficing/multi-core variants

- 30 minutes to solve each problem.
- What matters is only whether the problem was solved or not
- Optimal/Best solution has quality $Q^*$
- Planner finds a plan with quality $Q < Q^*$
- Quality ratio is $Q/Q^*$
- Objective function: maximize sum of quality ratios
Sequential **Agile** variant

- Satisficing solution *as soon as possible*
- Very **short amount of CPU time** available
- Domains and problems from **real-world applications**
- The aim to "simulate" planning techniques in a real environment
- Objective function: *minimize CPU time*
Sequential **Agile** variant

- 5 minutes to solve each problem.
- The **quality** of the resulting plans is **not important**
- What matters is only whether the problem was **solved or not** within 5 mins, and the **CPU time** required.
- **Minimum time required** by any planner is $T^*$
- Planner solves the problem **in time** $T$
- For **solved problem** gets the planner **score** $1/(1 + \log_{10}(T/T^*))$
- For **not solved problem** gets the planner **score of 0**
- Objective function: **maximize sum of scores over all problems**
Sequential Multi-core variant

- **Growing interest** in multi-core/parallel computation in the planning community
- **Different cores simultaneously** and/or with different threads on each core
- **No GPU** available
- **Only one computer with a number of cores** available (four cores expected)
Resources

- Demo problems
- **Plan Validator for PDDL**
  VAL: The Plan Validator
- **PDDL 3.1** - description
Call for **Domains**

- Negative preconditions and/or conditional effects encouraged
- Relation to real applications desirable
- Only **one entry per team** allowed
Some of **demo problems**

- Sokoban
- TSP
- Elevators
- Transport
The evaluation process

- Competitors will be given a set of representative domain/problem instances to test their planners on their own machines.

- Final version of planners will be run on the actual competition domains/problems unknown to the competitors till this time
Participation

- The focus is on data-collection and presentation, with interpretation of results being understated.
- The real goal is to make as much data as possible available to the community.
- All competitors must submit an abstract (max. 300 words) and a 4-page paper describing their planners.
- All source codes of planners will be public.
Learning Track

The Quality subtrack

- Domains using the plan quality evaluation from the deterministic track
- Comparison of learning versus non-learning planners
- Quality metric from the recent deterministic competitions
- Three awards: overall, basic solver, and best learner
Learning Track

The Quality subtrack

- The learning stage
- The domain definition
- The problem generator
- Domain-specific Control Knowledge
- Sets of training files
- The testing stage
Learning Track
The Integrated Execution subtrack

- Planner generates plans as part of a much larger system
- Learning and planning within the context of a simple execution loop
- Focus on fully observable, discrete, non-adversarial, deterministic, single-agent domains
- Awards: best overall learner, most adaptable learner, best anytime learner
Probabilistic Planning Track
Continuous

• Domains written in RDDL or RDDL2

• Examples:
  • Traffic Control
  • Mars Rover
Probabilistic Planning Track

Discrete

- Domains written in **RDDL** and various translations
- Examples:
  - Game of life
  - Elevators
  - Traffic
My motivation is

- To practice my skills in planning
- To solve declaratively described problems
- To try out existing tools
My intention

- Trying out of some existing planners
- Examination of currently used techniques
- Creation of my own basic planner
  - Sequential deterministic track
  - Satisfiable subtrack
- Support for core features
- Usage of some interesting techniques
Time complexity

- Using negative pre-/post-conditions
- **Existence** of a plan:
  - \textsc{EXPSPACE-}\textsc{c}
- **Existence** of a plan for given maximal makespan:
  - \textsc{NEXPTIME-}\textsc{c}
Existing techniques

- State/Plan space planning
- Planning with planning graph
- Forward search
- Backward search (lifted, strips)
- CSP, SAT
- Domain knowledge
- Abstraction, heuristics
Some preferred techniques

- Plan space planning
- Local changes
- Domain knowledge
- Abstraction
- CSP/SAT for some subproblems
Thank you for your attention

Questions && Answers

More information on:

ipc.icaps-conference.org
Sources

- **ICAPS Competitions** - webpage
  http://ipc.icaps-conference.org/

- **Fast-Forward Domain Collection** by Joerg Hoffmann
  http://fai.cs.uni-saarland.de/hoffmann/ff-domains.html

- **VAL: The Plan Validator**
  http://www.inf.kcl.ac.uk/research/groups/PLANNING/index.php?option=com_content&id=70&Itemid=77

- **Action description language** (ADL)

- Lectures on **Planning and Scheduling**
  http://ktiml.mff.cuni.cz/~bartak/planovani/