An Overview of the International Planning Competition Part 4: Getting Involved

Amanda Coles[1] Andrew Coles[1]
Álvaro Torralba[2] Florian Pommerening[3]

[1]King's College London

[2]Saarland University

[3]University of Basel, Switzerland

January 27, 2019

Write a Planner

Have an idea for a new technique?

Many tools available

- domains: `planning.domains`, `bitbucket.org/aibasel`
- translator: `fast-downward.org`
- planning framework: `fast-downward.org`
- validator: `github.com/KCL-Planning/VAL`, `github.com/patrikhaslum/INVAL`

Demo: Add a New Heuristic to Fast Downward

Submit a Planner

Want to submit your planner?

- different submission procedures over the years
- container technology used in 2018: Singularity

$\rightsquigarrow$ containerized versions of all 2018 participants available

Demo: Add a Singularity Script to Fast Downward

Organize an IPC Track

Interested in a track?

- Organize it!
- Don't wait for the next "classical" track.
- Get in touch
  - ICAPS competition liaison (Scott Sanner)
  - previous organizers like us (`ipc2018.bitbucket.io`)

Contribute to the IPC Workshop

IPC Workshop at ICAPS 2019

- result analyses
- track/rule suggestions
- opinion papers
- benchmarks
- metrics
- tools

Format

- 30/15/5 minutes presentations
- discussions

Thank you!
We hope to see you at the next IPC!

Demo Details (Get Fast Downward)

```
# install dependencies
sudo apt-get install cmake g++ mercurial make python

# clone Fast Downward
hg clone http://hg.fast-downward.org fast-downward
```

Demo Details (Create Heuristic)

```
# create a branch
cd fast-downward
hg branch ipc-seq-opt

# create your heuristic
cd src/search/heuristics
hg cp blind_search_heuristic.cc my_heuristic.cc
hg cp blind_search_heuristic.h my_heuristic.h
gedit my_heuristic.* ../DownwardFiles.cmake
cd ../../..

# Add to compiled files
gedit src/search/DownwardFiles.cmake

# Build and test
./build.py
/fast-downward.py misc/tests/benchmarks/gripper/* --search "astar(
```

Demo Details (Add Singularity Script and Commit)

```
# Add singularity script
rm -rf builds
gedit Singularity

# commit
hg commit -m "Added tutorial heuristic"
```

Demo Details (Entry in DownwardFiles.cmake)

```
fast_downward_plugin(
    NAME MY_HEURISTIC
    HELP "Tutorial demo heuristic"
    SOURCES
        heuristics/my_heuristic
    DEPENDS TASK_PROPERTIES
)
```

Demo Details (my_heuristic.h)

```
#ifndef HEURISTICS_MY_HEURISTIC_H
#define HEURISTICS_MY_HEURISTIC_H

#include "../heuristic.h"

namespace my_heuristic {
class MyHeuristic : public Heuristic {
    int max_value;
protected:
    virtual int compute_heuristic(const GlobalState &global_state)
public:
    MyHeuristic(const options::Options &opts);
};
}

#endif
```

Demo Details (my_heuristic.cc)

```
#include "my_heuristic.h"

#include "../global_state.h"
#include "../option_parser.h"
#include "../plugin.h"

#include "../task_utils/task_properties.h"

using namespace std;

namespace my_heuristic {
MyHeuristic::MyHeuristic(const Options &opts)
    : Heuristic(opts),
      max_value(opts.get<int>("max_value")) {
}

int MyHeuristic::compute_heuristic(const GlobalState &global_state) {
    State state = convert_global_state(global_state);
    int unsatisfied_goal_count = 0;

    for (FactProxy goal : task_proxy.get_goals()) {
        const VariableProxy var = goal.get_variable();
        if (state[var] != goal) {
            ++unsatisfied_goal_count;
            if (unsatisfied_goal_count == max_value) {
                break;
            }
        }
    }
    return unsatisfied_goal_count;
}
```

Demo Details (my_heuristic.cc, continued)

```
static shared_ptr<Heuristic> _parse(OptionParser &parser) {
    Heuristic::add_options_to_parser(parser);
    parser.add_option<int>(
        "max_value",
        "Maximal heuristic value (just to show how to add a parameter)",
        "infinity",
        Bounds("1", "infinity"));
    Options opts = parser.parse();
    if (parser.dry_run())
        return nullptr;
    else
        return make_shared<MyHeuristic>(opts);
}

static Plugin<Evaluator> _plugin("my_heuristic", _parse);
}
```

Demo Details (Singularity)

```
Bootstrap: docker
From: ubuntu

%setup
    REPO_ROOT=`dirname $SINGULARITY_BUILDDEF`
    cp -r $REPO_ROOT/ $SINGULARITY_ROOTFS/planner

%post
    apt-get update
    apt-get -y install cmake g++ make python
    cd /planner
    ./build.py

%runscript
    DOMAINFILE=$1
    PROBLEMFILE=$2
    PLANFILE=$3

    ## Call your planner.
    /planner/fast-downward.py \
        --plan-file $PLANFILE \
        $DOMAINFILE \
        $PROBLEMFILE \
        --search "astar(my_heuristic(max_value=3))"
```