Automatic Instance Generation for Classical Planning

Álvaro Torralba, Jendrik Seipp and Silvan Sievers
alto@cs.aau.dk, jendrik.seipp@liu.se, silvan.sievers@unibas.ch

**Evaluation in Classical Planning**

Usually, the evaluation in Planning follows the ICAPS/IPC way:
- Measure coverage.
- Time limit 5 or 30 minutes.
- Memory limit 2-8 GB.
- Use the benchmarks from the International Planning Competition.

**Why?**

Having a standard evaluation setting is beneficial:
- Reproducibility.
- Interpretability.
- Avoid hand picking results.

Benchmarks are an important part of evaluation in Planning.

But... The IPC benchmark set has some flaws

- Different number of instances per domain.
- Instance scaling: On some domains, IPC benchmarks do not show differences between planners even if they exist.

**Autoscale’21: The Benchmark Set**

**New set of benchmarks for Optimal and Satisficing Planning:**
- Uniform number of instances (30 instances)
- Includes almost all IPC STRIPS domains
- Optimization based on IPC’11, IPC’14 AND IPC’18 planners
- Useful to evaluate current and future planners.
- Example of Evaluation using Autoscale’14 (which didn’t use IPC’18 planners):

**Autoscale: An Automatic Tool to Select Instances**

**Principles:**
- Useful to evaluate current planners.
- Avoid bias.
- Keep the Spirit of the Domain.

**Rules:**

1. **Agnostic to individual Planner Performance:** Don’t consider the individual results of all planners available for the optimization (only the best and the worst per instance).

2. **Smooth Scaling:** The instance set should have:
   - Easy instances
   - Hard instances
   - Scale smoothly

3. **Parameter-based Selection:** Avoid selecting the random seed.

4. **Sequence-based Selection:** The parameter configurations can be organized in one or more sequences.

5. **User Constraints:** The domain designer specifies guidelines on which parameters to scale.

**Optimization Process**

- Generate candidate sequences with smooth scaling (using SMAC)
- Choose sub-sequences including easy, hard and diverse instances

**How to Use Autoscale to Generate Instance Sets?**

```bash
generator_command = "nomystery -l {locations} -p {packages} -n {edgefactor} -m {edgeweight} -c {constrainedness} -s {seed} -e 0"
p parameters = [
    LinearParam("locations", lower_b=3, upper_b=10, lower_m=0.1, upper_m=1),
    LinearParam("packages", lower_b=2, upper_b=20, lower_m=1),
    ConstantParam("edgefactor", 1.5),
    ConstantParam("edgeweight", 25),
    EnumParam("constrainedness", [1.1, 1.5, 2.0])
]
```

**Experiments**

How evaluate the quality of a benchmark set?
- Coverage range:
  - Some instances are solved by all planners
  - No planner solves all instances
- Comparisons: pairs (X,Y) of planners with different coverage

Comparison between Autoscale’14 and IPC:
- Instance Selection: 6 planners up to IPC’14
- Evaluation: 8 planners from IPC’18

**Conclusions**

1. A tool to automatically select instances which:
   - Create useful benchmarks
   - Is based on sequences of parameters
   - Avoids bias with respect of the planners used
   - Keeps the spirit of the domain


https://github.com/AI-Planning/autoscale
https://github.com/AI-Planning/autoscale-benchmarks