**Planning and Optimization**

G. Röger, T. Keller
G. Francès

University of Basel
Fall Semester 2018

# Classroom Exercise 1

*The files required for this exercise are in the directory* ***`classical/hands-on-1`*** *of the course repository (****`https://bitbucket.org/aibasel/planopt-hs18`****). Update your clone of the repository with* ***`hg pull -u`*** *to see the files. For the runs with Fast Downward, set a time limit of 1 minute and a memory limit of 2 GB. Using Linux, such limits can be set with* `ulimit -t 60` *and* `ulimit -v 2000000`*, respectively.*

**Exercise 1** (Running Fast Downward)

Play around with the Fast Downward planner:

(a) The directory `classical/hands-on-1/tile` contains a PDDL formulation of the 15-Puzzle (`puzzle.pddl`, `puzzle01.pddl`) and of the Weighted 15-Puzzle (`weight.pddl`, `weight01.pddl`). To run Fast Downward, use the script `fast-downward.py` with the corresponding domain and problem files, specifying the search algorithm and the heuristic. Example for the 15-Puzzle, greedy best first search and the FF heuristic:

```
./fast-downward/fast-downward.py tile/puzzle.pddl tile/puzzle01.pddl \
          --heuristic "h=ff()" --search "eager_greedy([h])"
```

Run Fast Downward on the 15-Puzzle and the Weighted 15-Puzzle, using greedy best first search and different heuristics:

- FF heuristic: `ff()`; additive heuristic: `add()`; blind heuristic: `blind()`

(b) Compare the results with respect to time, number of expanded and generated states, and solution quality.

**Exercise 2** (Glued 15-Puzzle)

Consider a modified version of the 15-Puzzle where some tiles are *glued* to their initial position. Tiles that are glued cannot be moved by any action. Modify the PDDL formulation of the domain file `puzzle.pddl` and the problem file `puzzle01.pddl` accordingly:

- Introduce an additional predicate `GLUED` in the domain file that indicates whether a tile is glued or not. Modify the action descriptions such that only tiles that are not glued can be moved.
- Modify the problem file such that tile 6 is glued.
- Run Fast Downward on the Glued 15-Puzzle with greedy best first search and the heuristics from Exercise 1. Compare the results with the results for the 15-Puzzle and the Weighted 15-Puzzle with respect to time, number of expanded and generated states, and solution quality.

**Exercise 3** (Cheating 15-Puzzle)

Consider another modified version of the 15-Puzzle where it is allowed to *cheat* in the sense that there are actions that

- allow to remove a tile from the frame (leaving the former position of the tile blank), and
- allow to reinsert a removed tile at any blank position.

Modify the PDDL formulation of the 15-Puzzle accordingly. Run Fast Downward on the Cheating 15-Puzzle with greedy best first search and the heuristics from Exercise 1. Compare the results with the results for the 15-Puzzle, the Weighted 15-Puzzle and the Glued 15-Puzzle with respect to time, number of expanded and generated states, and solution quality.

**Exercise 4** (Bonus Exercise: Package Delivery)

Consider the following package delivery problem:

- There is a set of *cities*, *trucks*, and *packages.*
- The cities are *connected* by a road network.
- A truck can *move* from one city to another along the roads.
- A package can be *loaded* into and *unloaded* from a truck.
- Every package has a *target location* where it should be delivered.

Model this domain in PDDL. Then model at least two different instances (e.g. different road networks, different target locations, different number of trucks) and find plans for them with Fast Downward.

*Please form groups of two students for classroom exercises.*