Planning and Optimization

M. Helmert, T. Keller
S. Eriksson, F. Pommerening, S. Sievers

University of Basel
Fall Semester 2019

Exercise Sheet G

Due: December 15, 2019

The files required for this exercise are in the directory `exercise-g` *of the course repository (*`https://bitbucket.org/aibasel/planopt-hs19`*). All paths are relative to this directory. Update your clone of the repository with* `hg pull -u` *to see the files.*

Exercise G.1 (7+3 marks)

(a) Explain the heuristics h^{pom} and h^{roc} introduced in the following paper.

> Felipe Trevizan, Sylvie Thiébaux, and Patrik Haslum. Occupation measure heuristics for probabilistic planning. In *Proc. ICAPS 2017*, pp. 306–315, 2017.

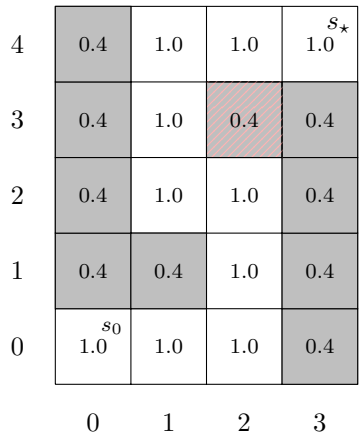
Use your own words and the notation from the lecture instead of the notation from the paper. Your explanation should include how the heuristics are computed, how they relate to each other and how they relate to operator-counting heuristics.

A good answer can be written in about 1 page.

(b) For operator-counting heuristics, the IP solution gives a stronger admissible heuristic estimate than the LP solution. Is this also true for h^{roc}? Explain why or provide a counterexample.

Exercise G.2 (9+1 marks)

Recall the SSP example from the lecture where an agent should move from an initial state to the goal state in a grid world. The figure below shows the grid with the initial state and the goal state. Possible moves are N, E, S, and W for the four directions, but actions are only applicable if they would not lead the agent out of the grid. In the goal state, no actions are applicable. The numbers in the grid indicate the probabilities p of an action to succeed, i.e., to move the agent in the intended direction. With probability $1 - p$, the agents stays where she is. The costs of applying an action is 1 in all cells except in the striped cell (2, 3), where it is 3.



In this exercise, you have to implement RTDP and LRTDP for the grid world problem. You only have to modify the file `rtdp/rtdp.py`. Please do not modify other files and only submit file `rtdp/rtdp.py`.

(a) Complete all functions with a TODO in file `rtdp/rtdp.py`.

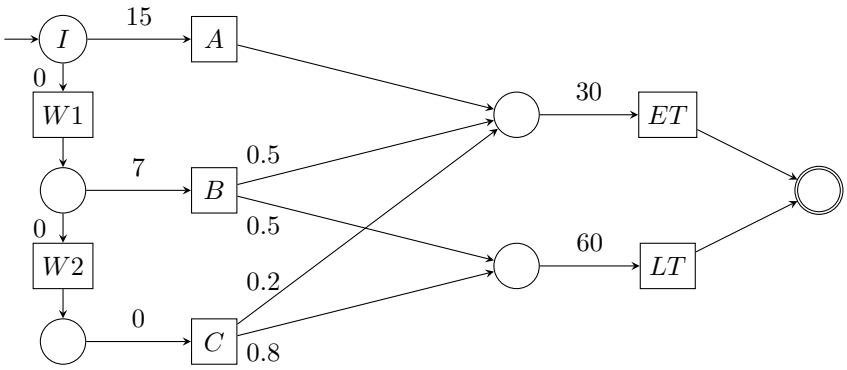
You can make yourself familiar with the problem representation by looking at `rtdp/instance.py`. Executing that file prints the probabilities and costs of all cells as well as the applicable actions for all states and the resulting successors. We recommend to implement the functions in this given order, since you should never need to use a function further down in the order in any of the earlier functions.

- `heuristic`: implement the Manhattan distance heuristic. For state (3, 4), the heuristic should be hard-coded to return 4 instead of the Manhattan distance.
- `compute_q_value`: compute the Q-value for a given state and action pair and a given value function.
- `compute_greedy_action_and_value`: for a given state and a given value function, compute the best applicable action and the resulting Q-value of that action.
- `get_greedy_policy`: for a given value function, compute a mapping from states to greedy actions.
- `sample_successor`: for a given state and action, sample a successor of all possible successors according to the probability.
- `perform_trial`: implement the trial function of RTDP.
- `residual`: compute the residual of a given state under a given value function.
- `check_solved`: implement the CheckSolved function of LRTDP.
- `visit`: implement the visit function of LRTDP.

(b) Run both algorithms using `rtdp/rtdp.py rtdp` and `rtdp/rtdp.py lrtdp`. How do the algorithms compare? Note that the algorithms are non-deterministic and you might want to execute them several times to get a clear picture.

Exercise G.3 (4 marks)

Assume you want to travel from the University to Lausen and want to minimize travel time. You can take tram A now, tram B in 7 minutes or tram C in 15 minutes (if you decide not to go now you can later still decide if you take the tram in 7 or 15 minutes). Depending on which tram you take you have a chance to miss your train and need to take the next one 30 minutes later. The travel time for tram and train (without waiting) is 30 minutes. For tram A, you will always catch the early train, for tram B you catch the early train with probability 0.5 and for tram C with probability 0.2. The following graph is one possible representation of this problem as an SSP:



Calculate the expected cost for choosing A or $W1$ in I when using Hindsight Optimization. What would the optimal policy do?

Exercise G.4 (6 marks)

Execute four iterations of MCTS for the task from Exercise G.2. Some steps of MCTS rely on policies and random sampling of outcomes during selection and simulation. To make the steps in the exercise interesting, use the following policies and choices.

- The tree policy always chooses the action with lowest current expected cost.
- The default policy always goes north if this action is applicable, or east if going north would leave the grid.
- When selecting actions in the selection phase, also use the default policy.
- Assume that outcomes are sampled in the following order
  (✓ means that the movement is successful, and ✗ means that the movement fails):
  - First iteration: ✓, ✓, ✓, ✓, ✗, ✗, ✗, ✓, ✗, ✗
  - Second iteration: ✗, ✗, ✗, ✗, ✓, ✗, ✗, ✓, ✓, ✓
  - Third iteration: ✗, ✓, ✓, ✗, ✓, ✓, ✗, ✗, ✗, ✗
  - Fourth iteration: ✓, ✗, ✗, ✗, ✗, ✓, ✗, ✓, ✗, ✓