Planning and Optimization

M. Helmert, G. Röger
F. Pommerening

University of Basel
Fall Semester 2017

Exercise Sheet F

Due: December 17, 2017

The files required for this exercise are in the directory `exercise-f` *of the course repository (*`https://bitbucket.org/aibasel/planopt-hs17`*). All paths are relative to this directory. Update your clone of the repository with* `hg pull -u` *to see the files.*

Exercise F.1 (4+3+2+2 marks)

(a) Consider the task from exercise E.3 and the three projections to a, b, and c. Compute an optimal cost partitioning and an optimal general cost partitioning for the abstractions. In each case, provide the cost partitioning as a table and the abstract transition systems of the projections. In the projections, annotate edges with their cost and states with their goal distances under the cost partitioning.

Discuss the differences between the two ways of partitioning the costs.

(b) Let V be a variable that is not mentioned in the goal of a task that is not trivially unsolvable. Prove that the projection to V cannot contribute in a cost partitioning with non-negative costs but it can contribute with general costs. Formally, this means showing the following two claims:

- Let $\mathcal{H}$ be a set of heuristics. Then the optimal non-negative cost partitioning over $\mathcal{H}$ is the same as the optimal non-negative cost partitioning over $\mathcal{H} \cup \{h^V\}$.
- There is at least one example of $\mathcal{H}$ and V where the optimal general cost partitioning over $\mathcal{H}$ is worse than the optimal general cost partitioning over $\mathcal{H} \cup \{h^V\}$.

Bonus exercise: find an unsolvable task where all projections to single variables are solvable but optimal general cost partitioning over the projections shows unsolvability.

(c) In the lecture, we have seen how to compute a uniform cost partitioning for a set of disjunctive action landmarks (chapter F2). Analogously, the set of disjunctive action landmarks which is built during the computation of the LM-cut heuristic induces a (non-uniform) cost partitioning. For a sequence $\langle L_1, \dots, L_n \rangle$ of disjunctive landmarks that is computed by LM-cut, formalize the induced cost partitioning $\langle cost_{L_1}, \dots, cost_{L_n} \rangle$ by LM-cut.

(d) Let $\mathcal{L}$ be a set of disjunctive action landmarks for a state s. Can there be a difference between the optimal non-negative and the optimal general cost partitioning of the landmark heuristics for landmarks in $\mathcal{L}$? Prove your answer.

Exercise F.2 (4+5+3+3 marks)

(a) Install Scip and its Python interface by following the instructions in the directory `scip`. Use it to implement operator-counting heuristics for the Python planner *pyperplan*. In the file `pyperplan/src/heuristics/heuristics/opcount.py` you can find an incomplete implementation. Finish it by following the comments in the code.

To use pyperplan with an A* search and a heuristic h on a task, use

`./pyperplan/src/pyperplan.py -s astar -H` h `/path/to/task`

where h is either `operatorcountingpho` (posthoc optimization constraints for all PDBs that project to 2 variables), `operatorcountinglm` (landmark constraints from LM-cut), or `operatorcountinglmpho` (an operator-counting LP that combines both of the constraints above).

Run your implementation of all three heuristics on the instances in the directory `logistics00` and compare the initial heuristic value, the optimal plan cost, and the number of expansions.

(b) Describe the following heuristics as potential heuristics by defining appropriate state features and a weight for each feature. When defining the features, make reasonable assumptions about the encoding of the problem and describe them as well.

- Use atomic features to encode the Manhattan distance heuristic in the sliding tile puzzle.
- Use atomic features to encode the "last move enhancement" of the Manhattan distance heuristic in the sliding tile puzzle. (We do not consider linear conflicts here, so ignore issues arising from a tile influencing both the last move and a linear conflict.)
- Use binary features to encode the "corner enhancement" of the Manhattan distance heuristic in the sliding tile puzzle. A binary feature $f_{X=x \wedge Y=y}$ is defined analogously to a atomic feature as

$$f_{X=x \wedge Y=y}(s) = \begin{cases} 1 & \text{if s[X] = x and s[Y] = y} \\ 0 & \text{otherwise.} \end{cases}$$

- Use atomic features to encode the goal-counting heuristic in an SAS$^+$ task.
- Use atomic features to encode the material value of a chess position.

You can find details about the sliding tile puzzle, the Manhattan distance heuristics and its enhancements in the following paper:

> Richard Korf and Larry Taylor. Finding optimal solutions to the twenty-four puzzle. In *Proc. AAAI 1996*, pp. 1202–1207, 1996.

For information on the material value of a chess position see for example: `https://en.wikipedia.org/wiki/Chess_piece_relative_value`

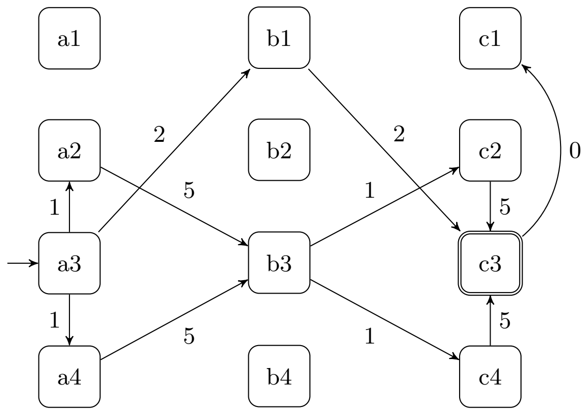
(c) Consider the linear constraints that characterize admissible and consistent atomic potential heuristics. Find a parametrized objective function for each of the following use cases. To be suitable as an objective function in an LP solver, the function should be linear in the weights and the number of its coefficients should be polynomial in the number of atoms (and the number of sample states where we use them).

- Maximize the heuristic value of the initial state.
- Maximize the sum of heuristic values for the states in a given set of sample states S.
- Maximize the average of heuristic values of all states (including unreachable states).

(d) Consider the planning task with two variables with domains $dom(v_1) = \{a, b, c\}$ and $dom(v_2) = \{1, 2, 3, 4\}$ shown below. States $\{v_1 \mapsto x, v_2 \mapsto y\}$ are shown as xy and every transition is caused by a separate operator with the cost of the operator shown next to the transition.

Find consistent and admissible atomic potential heuristics for this task that maximize the three objectives in exercise (c). (Use the sample set $S = \{b3, c2\}$ for the objective maximizing the sum of heuristics in a sample set.)

Compute the heuristic values for all states and compare the three heuristic functions.



Exercise F.3 (3+1 marks)

(a) In the summary of slides F5 we said that compilability is used to compare the power of heuristics. Discuss what this statement means, specifically the meaning and the limits of the word "power" in that sentence.

Here are some questions to get your discussion started (the exercise is about writing a general discussion, not about answering the specific questions):

- All heuristics can be compiled into merge-and-shrink heuristics so in one sense it is the most powerful heuristic; but on the other hand we have seen in exercise B.1 (b) that not all state-of-the-art planners use merge-and-shrink. Should all planners switch to using merge-and-shrink? If not, in what sense is merge-and-shrink the most powerful heuristic?
- Is h^3 always better than h^1?
- If a new class of heuristics is found to be compilable to be compilable into pattern database heuristics, what does this mean in regards to exercise D.3 (a)?
- What is the notion of compilability useful for?

(b) Show that PDB heuristics can be compiled into potential heuristics.

The exercise sheets can be submitted in groups of two students. Please provide both student names on the submission.