Planning and Optimization

M. Helmert, G. Röger
P. Ferber, T. Keller, S. Sievers

University of Basel
Fall Semester 2020

Exercise Sheet E

Due: November 29, 2020

Important: for submission, consult the rules at the end of the exercise. Non-adherence to the rules will lead to your submission not being corrected.

The files required for this exercise are in the directory `exercise-e` *of the course repository (*`https://github.com/aibasel-teaching/planopt-hs20`*). All paths are relative to this directory. Update your clone of the repository with* `git pull` *to see the files. In the virtual machine,* `/vagrant/plan-opt-hs20` *is the repository.*

Exercise E.1 (1+2+2 marks)(Lecture E2)

Consider the STRIPS task $\Pi = \langle V, I, O, \gamma \rangle$ with variable set $V = \{a, b, c, d, e, f, g\}$, initial state I with $I(a) = \top$ and $I(v) = \bot$ for all $v \in V \setminus \{a\}$, operator set $O = \{o_1, o_2, o_3, o_4, o_5, o_6\}$ and goal $\gamma = e$. All operators in O have cost 1 and are defined as follows:

$$
\begin{aligned}
o_1 &= \langle a, b \wedge \neg a \rangle \\
o_2 &= \langle a, c \rangle \\
o_3 &= \langle b, d \rangle \\
o_4 &= \langle c, d \rangle \\
o_5 &= \langle d \wedge g, e \wedge f \rangle \\
o_6 &= \langle \top, g \rangle
\end{aligned}
$$

(a) Provide the simplified relaxed task graph $sRTG(\Pi^+)$ in graphical form.

(b) Compute the set of causal fact landmarks with the fixed-point algorithm introduced in Chapter E2. You can annotate the nodes of your graph from (a) as in the lecture, but denote in which order you update the values of the nodes, and if you change the same node several times provide all intermediate values.

(c) Are the following two operator sets disjunctive action landmarks? Briefly justify your answer (no formal proof required):

- $O_1 = \{o_1, o_3\}$
- $O_2 = \{o_3, o_4\}$

Exercise E.2 (5 marks)(Lecture E3)

Consider the delete-free STRIPS planning task $\Pi^+ = \langle V, I, O, \delta \rangle$ with $V = \{i, a, b, c, d, e, g\}$, $I = \{i \mapsto \mathbf{T}\} \cup \{v \mapsto \mathbf{F} \mid v \in V \setminus \{i\}\}$, $\delta = g$, and $O = \{o_1, \dots, o_7\}$, where

$$
\begin{aligned}
o_1 &= \langle \{i\}, \{a\}, \{\}, 1 \rangle \\
o_2 &= \langle \{i\}, \{b\}, \{\}, 1 \rangle \\
o_3 &= \langle \{a\}, \{b, c\}, \{\}, 4 \rangle \\
o_4 &= \langle \{b\}, \{a, c\}, \{\}, 5 \rangle \\
o_5 &= \langle \{a, b\}, \{d\}, \{\}, 3 \rangle \\
o_6 &= \langle \{c, d\}, \{e\}, \{\}, 2 \rangle \\
o_7 &= \langle \{d, e\}, \{g\}, \{\}, 0 \rangle.
\end{aligned}
$$

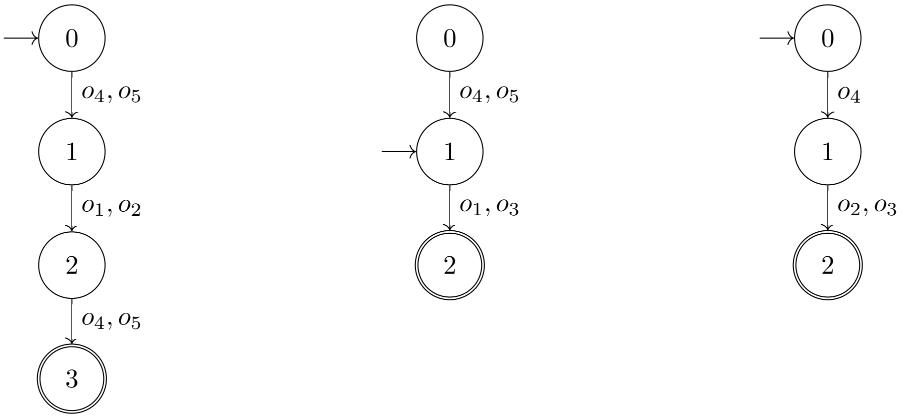
Compute $h^{\text{LM-cut}}(I)$ and provide all intermediate results in the same way they were given in the example of the lecture. Specifically, provide for each iteration (except the last):

- the justification graph with $h^{\max}$ annotations and marked goal zone,
- the cut,
- the cost of the cut, and
- the updated costs.

If multiple preconditions of an operator have the same maximal $h^{\max}$ value, the precondition choice function breaks ties in alphabetical order (i.e. for an operator $o = \langle\{a, b\}, \{c\}, \{\}, 2\rangle$ with $h^{\max}(a) = 3$ and $h^{\max}(b) = 3$, we have $pcf(o) = a$).

Exercise E.3 (1+1+1+2 marks)(Lecture E5)

Consider the planning task $\Pi = \langle V, I, O, \gamma\rangle$ with $V = \{a, b, c\}$, $dom(a) = \{0, 1, 2, 3\}$, $dom(b) = dom(c) = \{0, 1, 2\}$, $I = \{a \mapsto 0, b \mapsto 1, c \mapsto 0\}$, $O = \{o_1, o_2, o_3, o_4, o_5\}$, $cost(o_i) = 1$ for all $i \in \{1, 2, 3, 5\}$, $cost(o_4) = 2$, and $\gamma = (a = 3) \wedge (b = 2) \wedge (c = 2)$. The transition systems of the projections to variables are shown below: $\mathcal{T}^{\pi_{\{a\}}}$ on the left, $\mathcal{T}^{\pi_{\{b\}}}$ in the middle, $\mathcal{T}^{\pi_{\{c\}}}$ on the right.



(a) For abstraction heuristics $h^{\alpha_1}, \dots, h^{\alpha_n}$, the uniform cost partitioning is defined for all $o \in O$ and $1 \le i \le n$ by

$$cost_i(o) := \begin{cases} \frac{cost(o)}{n_o} & \text{if } o \text{ affects } \mathcal{T}^{\alpha_i} \\ 0 & \text{otherwise,} \end{cases}$$

where n_o is the number of abstractions α such that o affects $\mathcal{T}^{\alpha}$.

Compute the uniform cost partitioning for the three abstraction heuristics $h^{\{a\}}, h^{\{b\}}, h^{\{c\}}$ and provide the corresponding heuristic value in the initial state I.

(b) For abstraction heuristics $h^{\alpha_1}, \dots, h^{\alpha_n}$, a zero-one cost partitioning is defined for all $o \in O$ by $cost_i(o) := cost(o)$ for exactly one $i \in \{1, \dots, n\}$, and $cost_j(o) := 0$ for $j \neq i$. Compute a zero-one cost partitioning for the three abstraction heuristics $h^{\{a\}}, h^{\{b\}}, h^{\{c\}}$ which yields the maximal heuristic value in the initial state I among all zero-one cost partitionings. Provide the corresponding heuristic value in I.

(c) Compute the saturated cost paritionings for the three abstraction heuristics $h^{\{a\}}, h^{\{b\}}, h^{\{c\}}$ for the two following orders: $\langle h^{\{a\}}, h^{\{b\}}, h^{\{c\}}\rangle$ and $\langle h^{\{c\}}, h^{\{a\}}, h^{\{b\}}\rangle$. For both cost partitionings, provide both the cost and the minimal saturated cost functions for all three heuristics and provide the heuristic value of the resulting cost partitioning heuristic in I.

(d) Provide a cost partitioning that yields a higher heuristic value than the cost partitionings from (a), (b) and (c). What is the value of the corresponding heuristic in the initial state?

Exercise E.4 (4+2+1 marks)(Lecture E6)
Consider the following task $\Pi = \langle V, I, O, \gamma \rangle$ with

- Variables $V = \{a, b, c, d\}$ with $dom(a) = dom(b) = \{1, 2\}$, $dom(c) = \{1, 2, 3\}$, and $dom(d) = \{1, 2, 3, 4\}$,
- Initial state $I = \{a \mapsto 1, b \mapsto 1, c \mapsto 1, d \mapsto 1\}$,
- Operators $O = \{o_1, \dots, o_6\}$ where
  - $o_1 = \langle (a = 2) \land (b = 1) \land (c = 1), (a := 1) \land (b := 2) \land (c := 3) \rangle$, $cost(o_1) = 3$
  - $o_2 = \langle (a = 1), (a := 2) \rangle$, $cost(o_2) = 5$
  - $o_3 = \langle (b = 1) \land (d = 1), (b := 2) \land (d := 2) \rangle$, $cost(o_3) = 3$
  - $o_4 = \langle (c = 1) \land (d = 1), (c := 2) \land (d := 3) \rangle$, $cost(o_4) = 3$
  - $o_5 = \langle (c = 2) \land (d = 3), (c := 1) \land (d := 4) \rangle$, $cost(o_5) = 1$
  - $o_6 = \langle (c = 2) \land (d = 2), (c := 3) \land (d := 4) \rangle$, $cost(o_6) = 1$
- Goal $\gamma = (b = 2) \land (c = 3) \land (d = 4)$.

(a) Use SoPlex to compute an optimal non-negative cost partitioning of the four projections to a, b, c, and d. Please submit the result in three forms: (i) a file encoding the LP with clear names for variables and constraints, (ii) a table with one column per operator and one row per abstraction where each cell contains the cost of the operator in the abstraction, and (iii) the abstract transition systems where edges are annotated with the local cost function and states are annotated with their abstract goal distances. In the drawn transition system, do not include transitions caused by operators that have no effect on the abstraction (in the LP, these operators have to be included of course).

Instructions on how to install and use SoPlex are in the file `soplex-readme.txt`*.*

To reduce the work load of this exercise, we provided LaTeX source code for drawing the abstract transition systems in the directory `cost-partitioning` *which you can adapt for your solution. This directory also contains a Python file encoding the task, in case you want to create the LP file with a program. However, programming is not required for this exercise and you should not submit source code.*

(b) Repeat exercise (a) for a general cost partitioning.

(c) In exercises (a) and (b) the projection $\Pi^{\pi_{\{a\}}}$ always has a heuristic value of 0. Does this mean that it is safe to ignore this abstraction in the cost partitioning? Justify your answer.

Exercise E.5 (2+2 marks)(Lecture E6)

(a) Provide an unsolvable task where all projections to single variables are solvable but optimal non-negative cost partitioning over the projections shows unsolvability. Explain your example or why no such task exists.

(b) Provide an unsolvable task where all projections to single variables are solvable but optimal general cost partitioning over the projections shows unsolvability. Explain your example or why no such task exists.

Exercise E.6 (2+3+2 marks)(Lectures E7, E7, E8)
Consider the following task $\Pi = \langle V, I, O, \gamma \rangle$ with

- Variables $V = \{a, b, c\}$ with $dom(a) = dom(c) = \{1, 2, 3, 4, 5\}$, and $dom(b) = \{1, 2, 3, 4, 5, 6, 7\}$,
- Initial state $I = \{a \mapsto 1, b \mapsto 1, c \mapsto 1\}$,
- Operators $O = \{o_1, \ldots, o_{12}\}$ where
  - $o_1 = \langle (a = 1), (a := 2) \rangle$
  - $o_2 = \langle (a = 2) \wedge (b = 1) \wedge (c = 1), (a := 1) \wedge (b := 3) \rangle$
  - $o_3 = \langle (a = 1) \wedge (b = 4) \wedge (c = 1), (b := 2) \wedge (c := 2) \rangle$
  - $o_4 = \langle (a = 1) \wedge (b = 3) \wedge (c = 1), (b := 4) \rangle$
  - $o_5 = \langle (a = 2) \wedge (b = 2) \wedge (c = 2), (b := 4) \rangle$
  - $o_6 = \langle (a = 2) \wedge (b = 4) \wedge (c = 2), (b := 3) \rangle$
  - $o_7 = \langle (a = 2) \wedge (b = 3) \wedge (c = 2), (b := 6) \wedge (c := 4) \rangle$
  - $o_8 = \langle (a = 2) \wedge (b = 3) \wedge (c = 2), (a := 3) \wedge (b := 5) \rangle$
  - $o_9 = \langle (a = 2) \wedge (b = 6) \wedge (c = 2), (a := 4) \wedge (c := 3) \rangle$
  - $o_{10} = \langle (c = 4), (c := 2) \rangle$
  - $o_{11} = \langle (a = 3) \wedge (b = 5) \wedge (c = 4), (a := 5) \wedge (b := 7) \wedge (c := 5) \rangle$
  - $o_{12} = \langle (a = 4) \wedge (b = 6) \wedge (c = 3), (a := 5) \wedge (b := 7) \wedge (c := 5) \rangle$

  and $cost(o) = 1$ for all $o \in O$,
- Goal $\gamma = (a = 5) \wedge (b = 7) \wedge (c = 5)$.

(a) Provide the LP solved by the flow heuristic for I as an input file for SoPlex (see exercise E.4). Use each atom as the constraint name for its flow constraint so it is easy to see which constraint belongs to which atom. Then solve the LP and provide the objective value.

As for exercise E.4, Python code encoding the task is available in the directory `network-flow` but no programming is required.

(b) Draw the transition systems of the three projections to a, b, and c. For operators that do not mention the variable, include just one representative self-loop at the goal state to keep the transition system concise. Annotate each edge with the flow of the operator from exercise (a) and highlight edges with a non-zero value. What do you notice in the abstractions? Discuss your observations.

As for exercise E.4, LaTeX source code for drawing the abstract transition systems is available in the directory `network-flow`.

(c) The sets $L_1 = \{o_4, o_6\}$ and $L_2 = \{o_{10}, o_{12}\}$ are disjunctive action landmarks for Π. Provide the LP solved by the operator counting LP heuristic h^{LP}_{C} for I, where C contains the flow constraints from part (a) and the operator counting constraints induced by L_1 and L_2. Then solve the LP and provide the objective value.

Exercise E.7 (7 marks)(Lecture E8)

In this exercise, we want to draw a connection from linear programming to delete-relaxed planning tasks. In particular, the paper below describes an IP model for the delete relaxation. Explain the IP in detail, i.e., explain the meaning of all involved variables and constraints. Use your own words and the notation from the lecture instead of the notation from the paper. Also explain how the IP can be extended with operator counting constraints, again using notation from the lecture.

> Tatsuya Imai and Alex Fukunaga. A Practical, Integer-Linear Programming Model for the Delete-Relaxation in Cost-Optimal Planning. In *Proc. ECAI 2014*. pp. 459–464, 2014.

A good answer can be written in about 1 page.

Submission rules:

- Exercise sheets must be submitted in groups of two or three students. Please submit one single copy of the exercises per group (only one member of the group does the submission).
- Create a single PDF file (ending .pdf) for all non-programming exercises. If you want to submit handwritten parts, include their scans in the single PDF in a reasonable resolution, so that they are readable but the PDF size is not astronomically large. Put the names of all group members on top of the first page. Use page numbers or put your names on each page. Make sure your PDF has size A4 (fits the page size if printed on A4).
- For programming exercises, only create and submit those code textfiles required by the exercise. Put your names in a comment on top of each file. Make sure your code compiles and test it!
- For the submission, you can either upload the single PDF or prepare a ZIP file (ending .zip, .tar.gz or .tgz; not .rar or anything else) containing the single PDF and the code textfile(s) and nothing else. Please do not use directories within the ZIP, i.e., zip the files directly.
- Name all files without spaces.
- Only upload one submission per group. Do not upload several versions, i.e., if you need to resubmit, use the same file name again so that the previous submission is overwritten.