

## Planning and Optimization

M. Helmert, G. Röger  
R. Christen, P. Ferber, T. Keller

University of Basel  
Fall Semester 2022

### Exercise Sheet 11 Due: December 12, 2022

**Important:** For submission, consult the rules at the end of the exercise. Non-adherence to these rules might lead to a penalty in the form of a deduction of marks or, in the worst case, that your submission will not be corrected at all.

#### Exercise 11.1 (1+5+1 marks)

Consider the delete-free STRIPS planning task  $\Pi = \langle V, I, O, \gamma \rangle$  with

$$\begin{aligned}V &= \{i, a, b, c, d, e, g\}, \\I &= \{i \mapsto \mathbf{T}\} \cup \{v \mapsto \mathbf{F} \mid v \in V \setminus \{i\}\}, \\O &= \{o_1, \dots, o_7\}, \\o_1 &= \langle i, a, 2 \rangle, \\o_2 &= \langle i, b, 1 \rangle, \\o_3 &= \langle a, b \wedge c, 3 \rangle, \\o_4 &= \langle b, a \wedge c, 5 \rangle, \\o_5 &= \langle a \wedge b, d, 3 \rangle, \\o_6 &= \langle c \wedge d, e, 2 \rangle, \\o_7 &= \langle d, g, 0 \rangle, \\o_8 &= \langle e, g, 0 \rangle, \text{ and} \\ \gamma &= g.\end{aligned}$$

- (a) Provide two fact landmarks  $LM_1 \neq i$  and  $LM_2 \neq g$  for  $\Pi$  such that there is a greedy necessary ordering  $LM_1 \rightarrow_{\text{gn}} LM_2$ .
- (b) 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  $h^{\text{max}}$  annotations and marked goal zone,
  - the cut,
  - the cost of the cut, and
  - the updated costs.

Also provide the disjunctive action landmarks that are computed. If multiple preconditions of an operator have the same maximal  $h^{\text{max}}$  value, the precondition choice function breaks ties in alphabetical order (e.g., if  $h^{\text{max}}(a) = h^{\text{max}}(b) = 3$ , we have  $pcf(o_5) = a$ ).

- (c) Provide a minimum hitting set for the set of disjunctive action landmarks computed in part (b).

#### Exercise 11.2 (3 marks)

A manufacturer produces four different goods A, B, C, and D on 5 different machines  $M_1, M_2, M_3, M_4$ , and  $M_5$ . Each machine runs at most 200 hours per month. To produce 1 ton of a good, *all* machines run for their time listed in the row of that product. E.g. to produce 1 ton of A,  $M_1$  runs for 5 hours,  $M_3$  runs for 6 hours, and  $M_5$  runs for 8 hours.  $M_2$  and  $M_4$  are not used to produce A. Goods can be produced in parallel.

	$M_1$	$M_2$	$M_3$	$M_4$	$M_5$
A	5	0	6	0	8
B	3	6	4	3	0
C	4	5	3	3	4
D	4	2	0	2	3

Goods A and C are sold for 800 CHF per ton, good B is sold for 600 CHF per ton and good D for 500 CHF per ton. The maximum demand is 10 tons per month for goods A and B and 20 tons per month for goods C and D.

Formalize the problem as a linear problem (LP). Encode it as input to SoPlex and use SoPlex to solve it. Provide the solution found by SoPlex, i.e., the objective value and the values of all variables. Please also submit a single file encoding the LP with clear names for variables and constraints.

*Hint: You find instructions to install and to execute SoPlex, as well as to formalize LPs for SoPlex, in the directory **manufacturer** of the teaching repository.*

### Submission rules:

- Exercise sheets must be submitted in groups of two or three students. Please submit a 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. Use a file name that does not contain any spaces or special characters other than the underscore “\_”. If you want to submit handwritten solutions, include their scans in the single PDF. Make sure it is in a reasonable resolution so that it is readable, but ensure at the same time that the PDF size is not astronomically large. Put the names of all group members on top of the first page. Either use page numbers on all pages 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 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. Code that does not compile or which we cannot successfully execute will not be graded.
- For the submission: if the exercise sheet does not include programming exercises, simply upload the single PDF. If the exercise sheet includes programming exercises, upload 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. Do not use directories within the ZIP, i.e., zip the files directly. After creating your zip file and before submitting it, open the file and verify that it complies with these requirements.
- Do not upload several versions to ADAM, i.e., if you need to resubmit, use the same file name again so that the previous submission is overwritten.