Planning and Optimization

M. Helmert, G. Röger
C. Büchner, R. Christen, S. Dold

University of Basel
Fall Semester 2023

Exercise Sheet 9

Due: December 4, 2023

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 marked at all.

Exercise 9.1 (0.5+0.5+0.5+1.5+1 marks)

In this exercise we ask you to to conduct experiments with merge-and-shrink in Fast Downward. This is *not* a programming exercise, there is no need to change any of the source files. Run the provided configurations and discuss the results as indicated below.

Update the course repository (`/vagrant/planopt-hs23` in your course VM) with `$ git pull`. Navigate to the new directory `sheet09` which contains the current version of Fast Downward and some PDDL problem instances. Use the benchmark instances in the directory `freecell` for your experiments. Moreover, use a time limit of 2 minutes, which can be imposed by executing `$ ulimit -S -t 120` in the console after logging into the virtual machine.
Use the following command for your experiments:

```
$ ./fast-downward/fast-downward.py --alias * freecell/problem.pddl
```

where `*` is a placeholder of the name of the desired configuration. We always use the linear merge strategy but vary other aspects.

Name	Description
`ms-f-pres`	f-preserving shrink strategy
`ms-f-pres-label`	f-preserving shrink strategy with label reduction
`ms-bisim`	bisimulation shrink strategy
`ms-bisim-label`	bisimulation shrink strategy with label reduction

Exercises (a)–(c) ask for data from running the configurations in the above table. Provide the data in tables with one row per configuration and one column per problem instance. Note that running one configuration on a particular problem instance allows you to fill the corresponding cells in all parts (a)–(c).

(a) Provide the time required to compute the merge-and-shrink abstraction (`Merge-and-shrink algorithm runtime`).

(b) Provide the initial h-value (`Initial heuristic value for ...`) and add an additional row for the h^* value of the initial state (`Plan cost`) for all tasks where any configuration finds a solution.

(c) Provide the number of expanded states before the last f-layer (`Expanded until last jump`). (On the last f-layer, the number of expanded states only depends on tie-breaking which we don't want to include in our evaluation.)

(d) Compare the different configurations based on the data from the prior parts of this exercise. Discuss your observations in 3–4 sentences, taking into account your knowledge from the lecture. Consider in particular the effect of adding label reduction to the merge-and-shrink framework.

(e) You should notice some unexpected results in your data from parts (b) and (c) compared to the theory on bisimulation. What is unexpected about the data? Can you find an explanation for this unexpected behavior in the logging output?

Exercise 9.2 (0.5+1+1.5 marks)
Consider the STRIPS planning task $\Pi = \langle V, I, O, \gamma \rangle$ with

$$\begin{aligned}
V &= \{a, b, c\}, \\
I &= \{a \mapsto \mathbf{T}, b \mapsto \mathbf{T}, c \mapsto \mathbf{F}\}, \\
O &= \{o_1, o_2\}, \\
o_1 &= \langle a, \neg b \land c, 2 \rangle \\
o_2 &= \langle c, b, 1 \rangle, \\
\gamma &= a \land b \land c.
\end{aligned}$$

(a) Convert Π to a STRIPS planning task $\Pi' = \langle V', I', O', G' \rangle$ in set representation.

(b) Provide $R(A, O)$ for all $A \subseteq V$ with $1 \leq |A| \leq 2$.

(c) Compute $h^2(I, G)$ with the algorithm presented in Chapter F01 of the lecture. In each step, provide $h^2(I, A)$ for a subset $A \subseteq V$ of your choice that is relevant for the computation of $h^2(I, G)$.

Exercise 9.3 (1.5+1.5 marks)
Consider the task $\Pi = \langle V, I, O, G \rangle$ with the set notation $V = \{w, x, y, z\}$, $I = \{w, x\}$, $O = \{o_1, o_2\}$, $o_1 = \langle \{x\}, \{y\}, \{w\}, 1 \rangle$, $o_2 = \langle \{y\}, \{z\}, \emptyset, 1 \rangle$, and $G = \{x, y, z\}$.

(a) Provide the task $\Pi^2 = \langle V'', I'', O'', G'' \rangle$ that results from the Π^2 compilation of Π.

(b) Use Π^2 to show that there is an admissible heuristic h such that $h^*_\Pi(I) < h_{\Pi^2}(I^2)$.

Submission rules:

- Exercise sheets must be submitted in groups of 2–3 students. Create a team on ADAM including all members of your group and submit a single copy of the exercises per group.
- 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 text file(s) 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 text file(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.