Planning and Optimization

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

University of Basel
Fall Semester 2020

Exercise Sheet F

Due: December 06, 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-f` 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 F.1 (3+4+2+1 marks)(Lecture F2)

Push your Luck is a simple dice game where the player repeatedly rolls a single fair die to get rewards. The player can roll the die until she decides to collect her "accumulated" reward. The accumulated reward is defined as the product of all die outcomes *since the last game reset*, or 0, if no die has been rolled yet since that reset (the beginning of the game is also considered a game reset). The game "resets" when either the player collects the accumulated reward, or *the die shows a number that had already appeared since the last reset.*
As an example, imagine that after three rolls with a six-sided die the outcomes have been $1, 3, 4$. The player might decide to collect now, in which case her reward will be $1 \cdot 3 \cdot 4 = 12$, and the game will reset. Or she might be tempted to wait a bit, since perhaps she gets a 6 on the next roll, in which case she could collect a much more attractive reward of $1 \cdot 3 \cdot 4 \cdot 6 = 72$. On the other hand, waiting is risky: if she gets any outcome in $\{1, 3, 4\}$, then the game resets, but without her getting any reward. In any case, the player can continue playing *forever.*

(a) Formalize the game with an N-sided die as an MDP $\mathcal{M}_{N,\gamma} = \langle S, A, R, T, s_0, \gamma\rangle$ where N and the discount factor γ are parameters. For the set of states S, use the powerset of the set $\chi = \{1, \dots, N\}$, i.e., $S = 2^{\chi}$. The interpretation is that χ contains all possible die rolls and a state $s \in S$ contains exactly the die rolls obtained since the last reset.

(b) The file `mdps/push-your-luck.py` contains a skeleton of a script to generate the LP model for computing V^* and an optimal policy for $\mathcal{M}_{N,\gamma}$. Complete the script for your MDP model from (a).

The script uses PySCIPOpt, a Python interface to an optimization suite including SoPlex. Use `install-scip.sh` to install it and have a look at `scipdemo.py` for example code.

(c) Use your script from (b) to experiment with the game for a six-sided die ($N = 6$) and a discount factor $\gamma = 0.8$. Discuss the following questions:

What is the optimal state value $V^*(s_0)$ corresponding to the initial state? What is the optimal action to take in that state? What is the optimal action to take after having obtained the sequence of die outcomes $1, 2, 3$? And the sequence $4, 5, 6$? How do the above answers depend on the discount factor that you use? Play a bit with the factor and observe the differences, if any.

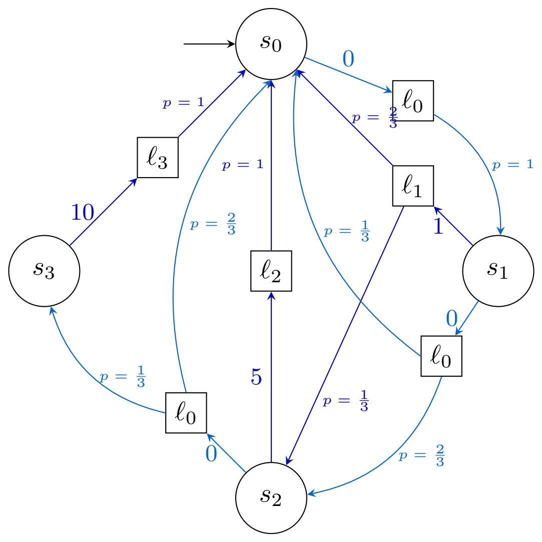
(d) Consider the variant of the game where the accumulated reward is defined as the *sum* instead of the product of all die outcomes since the last game reset, or 0, if no die has been rolled yet since that reset. How would this affect the optimal policy (independently of the chosen value for γ)? You don't have to recompute anything, but rather explain the effect of the changed reward function on the optimal values and, by that, on the optimal policy.

Exercise F.2 (5+5 marks)(Lecture F4)

Consider the discounted reward MDP $\mathcal{T} = \langle S, A, R, T, s_0, \gamma \rangle$ with

- $S = \{s_0, s_1, s_2, s_3\}$,
- $A = \{a_0, a_1, a_2, a_3\}$,
- $R(s, a_0) = 0$ for all $s \in S$
  $R(s, a_1) = 1$ for all $s \in S$
  $R(s, a_2) = 5$ for all $s \in S$
  $R(s, a_3) = 10$ for all $s \in S$,
- $T(s_0, a_0, s_1) = 1$
  $T(s_1, a_0, s_0) = \frac{1}{3}$ and $T(s_1, a_0, s_2) = \frac{2}{3}$
  $T(s_2, a_0, s_0) = \frac{2}{3}$ and $T(s_2, a_0, s_3) = \frac{1}{3}$
  $T(s_1, a_1, s_0) = \frac{2}{3}$ and $T(s_1, a_1, s_2) = \frac{1}{3}$
  $T(s_2, a_2, s_0) = 1$
  $T(s_3, a_3, s_0) = 1$
  $T(s, a, s') = 0$ for all other $s, s' \in S$ and $a \in A$
- $\gamma = 0.9$.

A graphical description of $\mathcal{T}$ can be seen below:



(a) Consider policy $\pi_0 = \{s_i \to a_i \mid i \in \{1, 2, 3, 4\}\}$ and perform *two iterations* of policy iteration, starting with initial policy π_0. Use iterative policy evaluation in the policy evaluation step and round state-values $\hat{V}_{\pi_0}(s_i)$ and $\hat{V}_{\pi_1}(s_i)$ for $i \in \{1, 2, 3, 4\}$ to two decimals (round the final result for each state-value, not intermediate results). Use $\epsilon = 0.235$ to terminate policy evaluation or after computing $\hat{V}^4_{\pi_i}(s_i)$, whichever comes first. Use $\hat{V}^0_{\pi_0}(s_0) = 10$, $\hat{V}^0_{\pi_0}(s_1) = 11$, $\hat{V}^0_{\pi_0}(s_2) = 13$, and $\hat{V}^0_{\pi_0}(s_3) = 18$ for iterative policy evaluation, and use the final state-values of the first iteration as initial state-values for the second iteration.

Provide all intermediate values of iterative policy evaluation under π_0 and π_1 in the two tables below (insert $\max_{s \in S} |V^i_{\pi_k}(s) - V^{i-1}_{\pi_k}(s)|$ in the last column) as well as the computed policies π_1 and π_2. Does policy iteration terminate after the second iteration or does it perform further iterations?

$\pi_0(s_0) =$ a_0 $\pi_0(s_1) =$ a_1 $\pi_0(s_2) =$ a_2 $\pi_0(s_3) =$ a_3

i	$\hat{V}^i_{\pi_0}(s_0)$	$\hat{V}^i_{\pi_0}(s_1)$	$\hat{V}^i_{\pi_0}(s_2)$	$\hat{V}^i_{\pi_0}(s_3)$	max. change
0	10	11	13	18	n/a
1					
2					
3					
4					

$\pi_1(s_0) =$ __________ $\pi_1(s_1) =$ __________ $\pi_1(s_2) =$ __________ $\pi_1(s_3) =$ __________

i	$\hat{V}^i_{\pi_1}(s_0)$	$\hat{V}^i_{\pi_1}(s_1)$	$\hat{V}^i_{\pi_1}(s_2)$	$\hat{V}^i_{\pi_1}(s_3)$	max. change
0					n/a
1					
2					
3					
4					

$\pi_2(s_0) =$ __________ $\pi_2(s_1) =$ __________ $\pi_2(s_2) =$ __________ $\pi_2(s_3) =$ __________

(b) Perform *five iterations* of value iteration with initial values $\hat{V}^0(s_0) = 10$ and $\hat{V}^0(s) = 0$ for $s \in \{s_1, s_2, s_3\}$. Provide the values for all states after each iteration in the table below, and provide the policies that result from the values after the 1st, 2nd, 3rd, 4th and 5th iteration. Would value iteration need to perform further iterations after the 4th iteration?

i	$\hat{V}^i(s_0)$	$\hat{V}^i(s_1)$	$\hat{V}^i(s_2)$	$\hat{V}^i(s_3)$
0	10	0	0	0
1				
2				
3				
4				
5				

$\pi_1(s_0) =$ __________ $\pi_1(s_1) =$ __________ $\pi_1(s_2) =$ __________ $\pi_1(s_3) =$ __________

$\pi_2(s_0) =$ __________ $\pi_2(s_1) =$ __________ $\pi_2(s_2) =$ __________ $\pi_2(s_3) =$ __________

$\pi_3(s_0) =$ __________ $\pi_3(s_1) =$ __________ $\pi_3(s_2) =$ __________ $\pi_3(s_3) =$ __________

$\pi_4(s_0) =$ __________ $\pi_4(s_1) =$ __________ $\pi_4(s_2) =$ __________ $\pi_4(s_3) =$ __________

$\pi_5(s_0) =$ __________ $\pi_5(s_1) =$ __________ $\pi_5(s_2) =$ __________ $\pi_5(s_3) =$ __________

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.