DPRL Assignment 2

Problem Description

Consider an inventory problem with 2 different products and an infinite time horizon

- At every timestep, the demand for the first product can either be 0 or 1, both with probability 0.5
- Similarly, the demand per timestep for the second product is also 0 or 1, both with probability 0.5

- Lost sales must be avoided, i.e. there should always be at least 1 item of each product in your inventory
- You have a limited storage capacity of 20 items per product

Note that the demand as well as the storage capacity for both products are fully independent of each other

- At every timestep, you could order any amount of the first and second product at the same time
- Every order has a fixed total cost of 5, regardless on the ordered amount of items/products
- Orders arrive at the end of the timestep (so can be sold the next timestep the earliest)

- The cost per timestep for holding a single item in stock is 1 for the first product and 2 for the second product

DPRL Assignment 2

Tasks

a) **Define** an appropriate state and action space for this problem. *Hint: The action space depends on the state*

For now, assume the order policy is fixed: Only when the inventory of one of the products is 1, you order such that the inventory level of both products becomes 5 (before any potential sales for that timestep take place).

b) **Describe** all possible transitions and their corresponding probabilities under the given policy.
c) **Simulate** the system for a long period under the given policy. **Report** the the long-run average costs.
d) **Compute** the limiting distribution using iteration under the given policy.
Use this distribution to **compute** and **report** the long-run average costs.
e) **Define** the average-cost Poisson equation under the given policy and **solve** it using value iteration.
Report the resulting long-run average costs.

Hint: part c, d & e should all result in the same long-run average costs

Now we stop using the fixed policy. We will instead be looking for an optimal policy instead.

f) **Define** the Bellman Equation and **solve** it using value iteration (minimizing the long-run average costs).
Find the optimal policy and **report** the corresponding long-run average costs.
g) **Interpret** your results and **characterize** the optimal policy.

How and what to submit

Hand in

- A report (.pdf):
  - Maximum 2 A4 pages, excluding an appendix with relevant figures/tables/screenshots.
  - OR maximum 1000 words with all figures/tables/screenshots inside your report.
- A separate file with Python source code (.py or .ipynb).

Make sure to

- Implement the algorithm in an efficient way, it should run very fast.
- Only use standard functions and packages (no MDC packages).
- Clearly describe all solution methods in your report:
  - Mathematically describe the methods that you coded.
  - Include all implementation choices, including initialization and stopping criteria.
- Comment on your findings, are they as expected?

Grading: 1 + 1 a + 0.5 b + 1 c + 1.5 d + 2 e + 2 f + 1 g