Exercise 8.1 (student presents)

A system without a disturbance is

$$x_{k+1} = x_k + u_k, \qquad x_k, u_k \in \mathbb{R}$$

and its initial state x_0. The cost to be minimized over all periods is

$$x_2^2 + u_0^2 + u_1^2.$$

There are no constraints for the control

a) What is the cost function $g_k(x_k, u_k)$ in periods $k = 0, 1, 2$?

b) Find the optimal control policy $u_k^* = \mu_k^*(x_k)$, $k = 0, 1$ with the DP-algorithm.

Solution

a) The cost function is piecewise:

$$g_2(x_2) = x_2^2, \qquad g_1(x_1, u_1) = u_1^2, \qquad g_0(x_0, u_0) = u_0^2$$

Explain why g_2 is independent of u_2!

b) $k = 2$:

$$J_2(x_2) = g_2(x_2) = x_2^2$$

$k = 1$:

$$J_1(x_1) = \min_{u_1} \Big[g_1(x_1, u_1) + J_2(x_2) \Big] = \min_{u_1} \Big[u_1^2 + J_2(x_2) \Big] = \min_{u_1} \Big[u_1^2 + (x_1 + u_1)^2 \Big]$$

Lets denote $L_1(x_1, u_1) = u_1^2 + (x_1 + u_1)^2$, which is continuous with regard to u_1 and x_2. The control that minimizes the cost-to-go function:

$$0 = \frac{\partial L_1}{\partial u_1} = 2u_1 + 2(x_1 + u_1)$$

out of which $u_1 = -x_1/2$. Because $\partial_{u_1}^2 L_1 > 0$, it is the minimum. Thus $\mu_1^*(x_1) = -x_1/2$ and $J_1(x_1) = x_1^2/2$.

$k = 0$:

$$J_0(x_0) = \min_{u_0}\left[u_0^2 + J_1(x_1)\right] = \min_{u_0}\left[u_0^2 + 0.5(x_0+u_0)^2\right].$$

$$0 = \frac{\partial L_0}{\partial u_0} = 2u_0 + (x_0 + u_0)$$

out of which $u_0 = -x_0/3$. It is the minimum, because $\partial^2_{u_0} L_0 > 0$. Thus $\mu_0^*(x_0) = -x_0/3$ and $J_0(x_0) = x_0^2/3$.

Now we can also solve u_1^* and x_2^* as functions of the initial state:

$$x_1^* = x_0 + u_0^* = x_0 - \frac{1}{3}x_0 = \frac{2}{3}x_0$$

$$u_1^* = -\frac{1}{2}x_1^* = -\frac{1}{2}\frac{2}{3}x_0 = -\frac{1}{3}x_0$$

$$x_2^* = x_1^* + u_1^* = \frac{2}{3}x_0 - \frac{1}{3}x_0 = \frac{1}{3}x_0$$

The cost of the optimal solution with the initial state x_0 is:

$$x_2^2 + u_0^2 + u_1^2 = \left(\frac{1}{3}x_0\right)^2 + \left(-\frac{1}{3}x_0\right)^2 + \left(-\frac{1}{3}x_0\right)^2 = \frac{1}{3}x_0^2$$

which is the same as $J_0(x_0)$.

Exercise 8.2 (solved in class)

Lets restrict the control u_k in Exercise 8.1 to attain only the values -1 and 1. Find the optimal control policy $u_k^* = \mu_k^*(x_k)$, $k = 0, 1$ with the DP-algorithm. Hint: the solution is piecewise.

Solution

$k = 2$:

$$J_2(x_2) = x_2^2.$$

$k = 1$:

$$\begin{aligned} J_1(x_1) &= \min_{u_1}\left[u_1^2 + J_2(x_2)\right] = 1 + \min_{u_1}\left[(x_1+u_1)^2\right] \\ &= 2 + x_1^2 + 2\cdot\min_{u_1}[x_1 u_1] \end{aligned}$$

What is the minimizing control u_1^*? It depends on the sign of x_1:

$$\begin{aligned} \mu_1^*(x_1) &= -\mathrm{sgn}(x_1) \qquad \forall x_1 \in \mathbb{R} \\ J_1(x_1) &= 1 + (1 - |x_1|)^2 = 2 + x_1^2 - 2|x_1| \end{aligned}$$

Here $\mathrm{sgn}(x_1) = 1$ when $x_1 \geq 0$, otherwise -1.

$k = 0$:

$$\begin{aligned} J_0(x_0) &= \min_{u_0} \left[u_0^2 + J_1(x_1) \right] = \min_{u_0} \left[u_0^2 + 1 + (1 - |x_0 + u_0|)^2 \right] \\ &= 2 + \min_{u_0} \left[(1 - |x_0 + u_0|)^2 \right] \\ &= 4 + x_0^2 + 2 \cdot \min_{u_0} [x_0 u_0 - |x_0 + u_0|] \end{aligned}$$

Separate the state variable into intervals:

$$\mu_0^*(x_0) = \begin{cases} 1, & x_0 < -1 \\ \pm 1, & -1 \leq x_0 \leq 1 \\ -1, & x_0 > 1. \end{cases}$$

The optimal cost is

$$J_0(x_0) = \begin{cases} 2 + x_0^2, & x_0 \in [-1, 1] \\ 6 + x_0^2 - 4|x_0|, & \text{else.} \end{cases}$$

Exercise 8.3 (teacher demo)

According to the application example in Lecture 2 your task is to to find the optimal alignment for the two DNA-sequences:

(1) G A A T T C A G T T A

(2) G G A T C G A

Lets assume that in evolution there are two types of mutations that happen more than other; C $\rightarrow$ T and T $\rightarrow$ C. Unlike the example in the lecture slides $S_{i,j} = 1$, if in sequence 1 in the place i is the structural unit T or C and in sequence 2 in the place j is the structural unit C or T, otherwise in a non-matching case $S_{i,j} = 0$.

Create a scoring matrix and using the DP-algorithm find the optimal alignment for sequences (1) and (2).

You can use the Matlab-codes *scoring.m* and *nwmatrix.m* and edit them.

Solution

Lets edit the Matlab-code scoring.m so that there are two more elseif-conditions:

```
elseif (strcmp(char1,'C') && strcmp(char2,'T'))
  S = 1;
elseif (strcmp(char1,'T') && strcmp(char2,'C'))
  S = 1;
```

The scoring matrix is

	G	A	A	T	T	C	A	G	T	T	A
G	1	1	1	1	1	1	1	1	1	1	1
G	1	1	1	1	1	1	1	2	2	2	2
A	1	2	2	2	2	2	2	2	2	2	3
T	1	2	2	3	3	3	3	3	3	3	3
C	1	2	2	3	4	4	4	4	4	4	4
G	1	2	2	3	4	4	4	5	5	5	5
A	1	2	3	3	4	4	5	5	5	5	6

One optimal alignment is:

```
(1) G A A T T C A G T T A
(2) G G A T C - - G - - A
```

Exercise 8.4 (self-study, answer only revealed after homework deadline)

Your task is to hire a handyman out of N candidates. The handymen arrive one by one for an interview to your office in a random order. In the interview you will find out how good the handyman is compared to those you have already interviewed. In the end of the interview, you can choose either to hire or reject the candidate. If you reject him, the next candidate will step into your office. If you hire him, you won't interview any more candidates. If you rejected all candidates before, so that the last candidate is in your office, you have to hire him.

Lets assume, that the handymen can be ordered by competence, so that no two handymen can be equally good. You will score a point, if you hire the best handyman, otherwise your utility is zero. Solve the optimal hiring policy with the DP-algorithm, when $N = 2, 3, 4$.

Exercise 8.5 (homework)

The Ombudsman for Equality has decided, that the problem of hiring a handyman should be changed into the problem of hiring a handyperson. Now a company faces the task to hire a handyperson. The pipes in the offices of the company are leaking so terribly that it doesn't matter to them whether the hired handyperson is the best or second best of the candidates. Thus, the problem is the same as in Exercise 8.4, expect for that now you score a point for hiring the best or the second-best handyperson.

Solve the optimal hiring policy with the DP-algorithm, for $N = 2, 3, 4$. Additionally, as a bonus problem (which doesn't affect the grading of your solution) solve what happens when $N \to \infty$.